

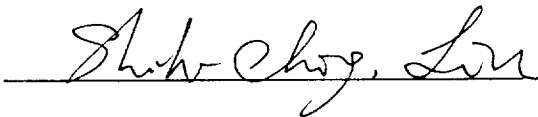
**A METHOD TO DETERMINE THE KINEMATICS  
OF THE LOWER LIMBS OF A SUBJECT PEDALING  
A BICYCLE USING ENCODERS AND ACCELEROMETERS**

A THESIS

SUBMITTED ON THE TWENTY FIRST DAY OF APRIL, 1994  
TO THE DEPARTMENT OF MECHANICAL ENGINEERING OF TULANE  
UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

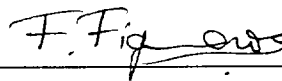
FOR THE DEGREE OF  
MASTER OF SCIENCE IN ENGINEERING

BY



Shih - Ching Liu

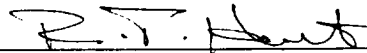
COMMITTEE APPROVAL:



Dr. J. F. Figueroa, Chair

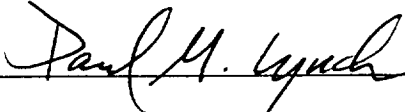


Dr. P. M. Lynch



Dr. R. T. Hart

DEPARTMENT APPROVAL :



Dr. P. M. Lynch, Head

Department of Mechanical

Engineering

N95-21866

Unclass

G3/54 0039174

(NASA-CR-197401) A METHOD TO  
DETERMINE THE KINEMATICS OF THE  
LOWER LIMBS OF A SUBJECT PEDALING A  
BICYCLE USING ENCODERS AND  
ACCELEROMETERS M.S. Thesis (Tulane  
Univ.) 68 p

## **ABSTRACT**

The goal of this research was to determine kinematic parameters of the lower limbs of a subject pedaling a bicycle. An existing measurement system was used as the basis to develop the model to determine position and acceleration of the limbs. The system consists of an ergometer instrumented to provide position of the pedal (foot), accelerometers to be attached to the lower limbs to measure accelerations, a recorder used for filtering, and a computer instrumented with an A/D board and a decoder board. The system is designed to read and record data from accelerometers and encoders. Software has been developed for data collection, analysis and presentation. Based on the measurement system, a two dimensional analytical model has been developed to determine configuration (position, orientation) and kinematics (velocities, accelerations).

The model has been implemented in software and verified by simulation. An error analysis to determine the system's accuracy shows that the expected error is well within the specifications of practical applications. When the physical hardware is completed, NASA researchers hope to use the system developed to determine forces exerted by muscles and forces at articulations. This data will be useful in the development of countermeasures to minimize bone loss experienced by astronauts in micro gravity conditions.

## **ACKNOWLEDGEMENT**

The author thanks his advisor, Dr. J F. Figueroa for his constant guidance and encouragement during this research. The author also thanks his advisor for the support in writing the thesis and for serving as chair of the committee.

The author would like to thank Dr. P. M. Lynch and Dr. R. T. Hart who are members of the committee. In addition, the author would like to thank his fellow graduate students, Ajay and Thomas, and his roommate, Bennett, for their constant encouragement and companionship.

Finally, the author is grateful to NASA and the American Association of Engineering Education for funding this research.

## TABLE OF CONTENTS

1 INTRODUCTION.....	1
2 BACKGROUND.....	4
3 PROPOSED SYSTEM.....	6
3.1 Introduction .....	6
3.2 Method .....	6
3.3 Kinematic equations.....	7
3.3.1 Determination of angular velocity and acceleration.....	8
3.3.2 Determination of angular position.....	12
3.3.3 Determination of linear acceleration.....	15
3.4 Error analysis .....	16
3.4.1 Analysis of error propagation.....	16
3.4.2 Numerical demonstration.....	23
4 SOFTWARE .....	28
4.1 Introduction .....	28
4.2 Data acquisition programs in C++ .....	28
4.2.1 Storage of data .....	29
4.2.2 Synchronization with encoders .....	31
4.2.3 Checking the D/A system .....	37
4.2.4 Discussion .....	37

4.3 The MATLAB program for kinematics.....	38
4.4 Simulation .....	39
5 CONCLUSIONS .....	42
6 RECOMMENDATIONS .....	44
LITERATURE CITED .....	45
APPENDIX A.1 Inputs to TEST.M .....	47
APPENDIX A.2 Accelerometer values produced by TEST.M.....	49
APPENDIX A.3 The results of simulation.....	51
APPENDIX B.1 KINET.M.....	53
APPENDIX B.2 TEST.M.....	60

## LIST OF FIGURES

Figure 1.1 Hardware connections during the experiment .....	2
Figure 3.1 Frames and parameters for the kinematic analysis.....	7
Figure 3.2 Effect of gravity on the accelerometer.....	9
Figure 3.3 Five-bar linkage system and position vectors on the calf section.....	24
Figure 3.4 Dependency of the error of the calf angle on % accelerometer error.....	26
Figure 3.5 Dependency of the error of the calf angle on the separation between accelerometers.....	27
Figure 4.1 Flowchart of SCANOPF.CPP .....	32
Figure 4.2 Flowchart of SCANOPF2.CPP.....	33
Figure 4.3 Flowchart of SCANOPB.CPP.....	34
Figure 4.4 Flowchart of SCANOPB2.CPP.....	35
Figure 4.5 Flowchart of SCANSTAR.CPP.....	36
Figure 4.6 Flowchart of KINET.M.....	39

## LIST OF TABLES

Table 4.1 Differences inside the data acquisition loop among the five programs.....	30
---	----

## INTRODUCTION

Decrease in muscle loading and external loading of bone during weightlessness in space can result in cancellous bone loss of 1% per month in the lower extremities and 2% per month in the calcaneus (Figuerola, 1993). Therefore bone loss is a serious problem encountered by astronauts that must remain in micro gravity conditions for the duration of a mission. It is hypothesized that loading bone appropriately during exercise may prevent bone loss (Figuerola, 1993). To minimize bone loss, NASA scientists are considering development of exercise countermeasures. This development involves definition of exercises and doses that will stress the bone so as to minimize bone loss on subjects participating in a bedrest study. The bedrest model is used to simulate micro gravity conditions. Loads on exercises that are considered effective must be quantized to be compared with loads measured during exercise in space. Thus, information about the kinematics, dynamics and forces exerted by a particular muscle or muscle group during exercise is necessary to quantify bone loading for the development of exercise countermeasures.

This thesis describes a system to determine the kinematic parameters of the lower limbs during exercise. The system elements were selected based upon specification from NASA. It consists of instrumentation, sensors and methodologies necessary to support the determination of kinematic and dynamic information. It encompasses a combination of small, light, and robust sensors, suitable for use in the confined environment of a space vehicle.

The research included refurbishing and improvement of existing hardware, development

of software, and development of a two dimensional model to determine kinematic parameters of the lower limbs of a subject riding an ergometer.

The equipment used includes an ergometer, a 28-channel recorder (XR-9000 Cassette Data Recorder, Teac Corp., Tokyo, Japan), accelerometers (EGAXT-10, Entran Devices Inc., Fairfield, N.J.), three rotary encoders, a 486 class personal computer fitted with two cards: (1) a three-channel decoder card (Model 5312 board, Technology Inc., Minneapolis, MN) to read the encoder information and (2) a 64 channel A/D board (AT-MIO-64F-5, National Inc., Austin, Texas) to digitize the data from the accelerometers.

Accelerometers are attached to the calf and thigh sections during exercise. Signals from the accelerometers are amplified before recording. Filtering is done using the recorder. The angular positions of the crank and the pedals may be measured by encoders (M20051221031, Dynapar Corp., Gurnee, IL) installed at the joints of the crank and pedals of the ergometer. The recorder performs anti-aliasing filtering. Figure 1.1 shows the hardware connections.

Data from the accelerometers and encoders are synchronously recorded. Data from

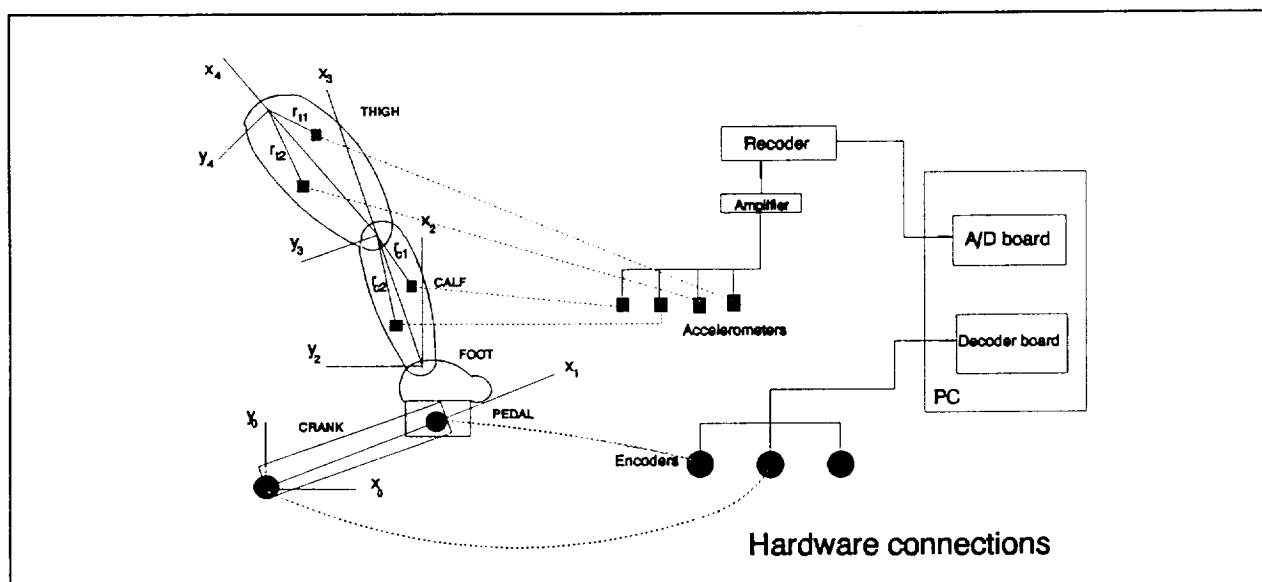


Figure 1.1



encoders is converted to angular positions, angular velocities, and angular accelerations. Data from the accelerometers is used in the kinematic analysis. The A/D board used in the research is the AT-MIO-64F-5 board produced by National Instruments Inc. It is a multifunction analog, digital, and timing I/O board for the personal computer. The board has a 5  $\mu$ sec, 12-bit sampling ADC that can monitor a single input channel, or scan through the 64 single-ended or 32 differential channels. NI-DAQ application programming software was used to drive the AT-MIO-64F-5 board.

A novel two dimensional analytical model was developed, which uses data from the above system as input to determine kinematic parameters. The model has been implemented in a computer program, KINET.M. The kinematic parameters are needed to determine bone and muscle loading, but this thesis does not include determination of loads.

Section Two describes the theoretical background of this research. Section Three gives a complete description of the theoretical model for kinematics. The error of the angular positions of the lower limbs resulting from the precision errors of the accelerometer values is also analyzed. Section Four describes the software of the proposed system. The last two sections include conclusions and recommendations.

## BACKGROUND

According to Wolff's law, living bone changes according to stress and strain conditions on the bone (Fung, 1919). The mechanism of bone remodeling is linked to strain or change in bone dimensions caused by stress applied to it. To increase bone strength, stress must be applied to produce sufficient strain. To develop exercise countermeasures that produce bone stress adequate to minimize bone loss in micro gravity conditions, it is necessary to quantify the loads during exercise.

Generally, forces and torques exerted at joints are measured by use of the equations of motion (Newton-Euler Law of Motion) of the body parts of interest (Figueroa, 1993). This results in a set of equations that relate the torques and forces applied by muscles, and by neighboring bone sections at the joints (Redfield and Hull, 1986; Anderson et. al., 1993; Yang et. al., 1993; Abdel-Rahman and Samir, 1993; Verstraete, 1991; Ericson et. al., 1985; Harrison et. al., 1986). To use the equations of motion, the measurement of kinematic parameters such as accelerations, positions, and orientations of the body parts is required. This is the topic addressed by this thesis.

Popular methods to measure the kinematic parameters during exercise use camera/light systems (Ericson et. al., 1985; Harrison et. al., 1986). These methods are accurate, but the instrumentation is bulky and requires large spaces, and the procedures to install and/or calibrate are time consuming. The systems also require unobstructed line of sight between cameras and lights, and are generally expensive. Another method, to describe the kinematics of the leg in

the sagittal plane only, assumes that the hip remains static, and applies kinematics of a five-bar linkage (Redfield and Hull, 1986A, Redfield and Hull, 1986B). The system developed uses classical engineering methods, more in line with the requirements put forth by NASA (compact, transferable to a space environment, easy to install and operate), and uses accelerometers and position sensing instruments. The method described in this thesis is unique in that although it uses accelerometers, position is not obtained by integration. This avoids the usual accumulation of errors characteristic of methods that integrate measured accelerations to determine position.

## **THE PROPOSED SYSTEM**

### **3.1 INTRODUCTION**

The proposed system uses a combination of accelerometers and rotary digital encoders. After passing through anti-aliasing filters, measurements made by these sensors are read into a computer with data acquisition cards: (1) AT-MIO-64F-5, National Inc., Austin, Texas and (2) Model 5312 board, Technology Inc., Minneapolis, MN; and then those measurements are used to determine the configuration and kinematics.

Section 3.2, Methods, and Section 3.3, Kinematic Equations, describe the measurement system and the methodology. Section 3.4 describes an analysis of error propagation from the accelerometer values to the angular position of the body section.

### **3.2 METHODS**

To apply the Newton-Euler Equations of Motion, the acceleration of the center of mass must be first determined. To obtain acceleration of the center of mass, miniature accelerometers are used to measure the acceleration vector of two points (for motion in the sagittal plane). The acceleration vector will be expressed with respect to a coordinate system attached to the member. The orientation of this coordinate system will be determined using the acceleration of a known point in both coordinate systems (a base coordinate system or the inertial frame, and the one attached to the member). The relative rotation of the two coordinate systems will be determined by the equations that relate the orthogonal components of the same acceleration

vector expressed in the two coordinated systems rotated with respect to each other.

Using accelerometers on Earth requires that the component of the acceleration of gravity along the direction of the accelerometer axis should be included as part of the measured acceleration. This component will be considered in the development of the model described in the next section.

### 3.3 KINEMATIC EQUATIONS

The formulation assumes that the acceleration of one point of the member of interest is known, and that accelerometers placed on the member measure acceleration components along the directions of the frame attached to the member. For example, Figure 3.1 shows the thigh, calf, and foot projections onto the sagittal plane. The position and orientation of the foot are measured using optical rotary encoders attached to the crank and pedal rotation axes. Angular velocities and angular accelerations of the crank and pedals can thus be known.

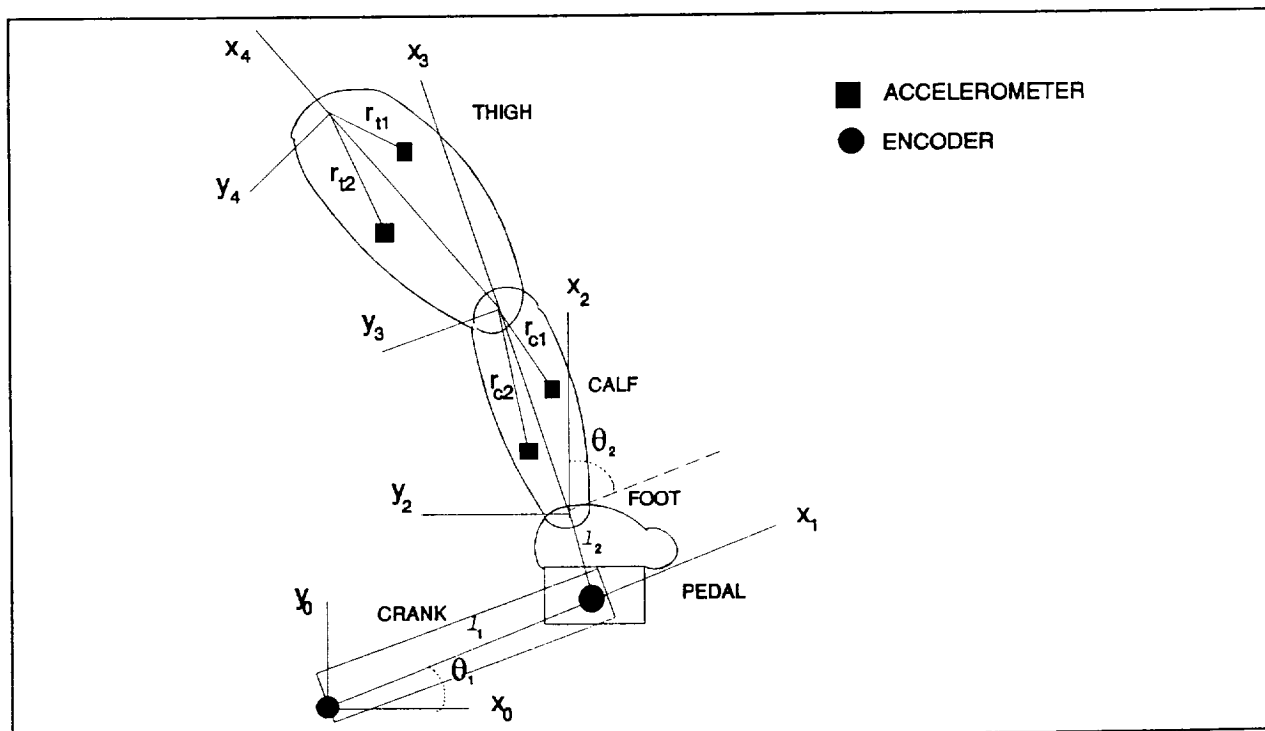


Figure 3.1

### 3.3.1 Determination of angular velocity and acceleration

The acceleration of the foot joint with respect to the inertial frame, denoted  ${}^0\mathbf{a}_2$ , can be determined using Equation 3.1. In this equation  $l_1$  is the length of the crank,  $l_2$  is the length of the pedal/foot,  $\theta_1$  is the angular position of the crank w.r.t the inertial frame,  $\theta_2$  is the relative angular position of the pedal with respect to the crank joint,  $\omega_1$  and  $\alpha_1$  are the angular velocity and acceleration of the crank,  $\omega_2$  and  $\alpha_2$  are the relative angular velocity and acceleration of the foot with respect to the crank,  $C_1$  is  $\cos(\theta_1)$ ,  $S_1$  is  $\sin(\theta_1)$ ,  $C_{12}$  is  $\cos(\theta_1 + \theta_2)$  and so on.

$${}^0\mathbf{a}_2 = \begin{bmatrix} -l_1 S_{12} & -l_2 S_{12} \\ l_1 C_1 & l_2 C_{12} \end{bmatrix} \begin{bmatrix} \alpha_1^2 \\ \alpha_1^2 + \alpha_2^2 \end{bmatrix} - \begin{bmatrix} l_1 C_1 & l_2 C_{12} \\ l_1 S_1 & l_2 S_{12} \end{bmatrix} \begin{bmatrix} (\omega_1)^2 \\ (\omega_1 + \omega_2)^2 \end{bmatrix} \quad (3.1)$$

Two accelerometers are attached to the calf; and another two to the thigh. The accelerations measured by these accelerometers are used to determine the angular velocities, angular accelerations, and linear accelerations of the centers of gravity of these sections. Since the accelerometers are used on Earth, a  $1g$  value in the vertical direction is added to the actual accelerations. As shown below, the gravity component increases the complexity of the method for determining the angular positions of the calf and thigh sections.

The linear acceleration of the knee joint with respect to Frame 3 is denoted  ${}^3\mathbf{a}_3$ . The linear acceleration of accelerometer 1 attached to the calf with respect to Frame 3, is denoted  ${}^3\mathbf{a}_{c1}$ , and can be expressed as follows:

$${}^3\mathbf{a}_{c1} = {}^3\mathbf{a}_3 + \alpha_3 \times {}^3\mathbf{r}_{c1} + \omega_3 \times (\omega_3 \times {}^3\mathbf{r}_{c1}) \quad (3.2)$$

where  $\alpha_3$  denotes angular acceleration of the calf,  $\omega_3$  angular velocity of the calf, and  ${}^3\mathbf{r}_{c1}$

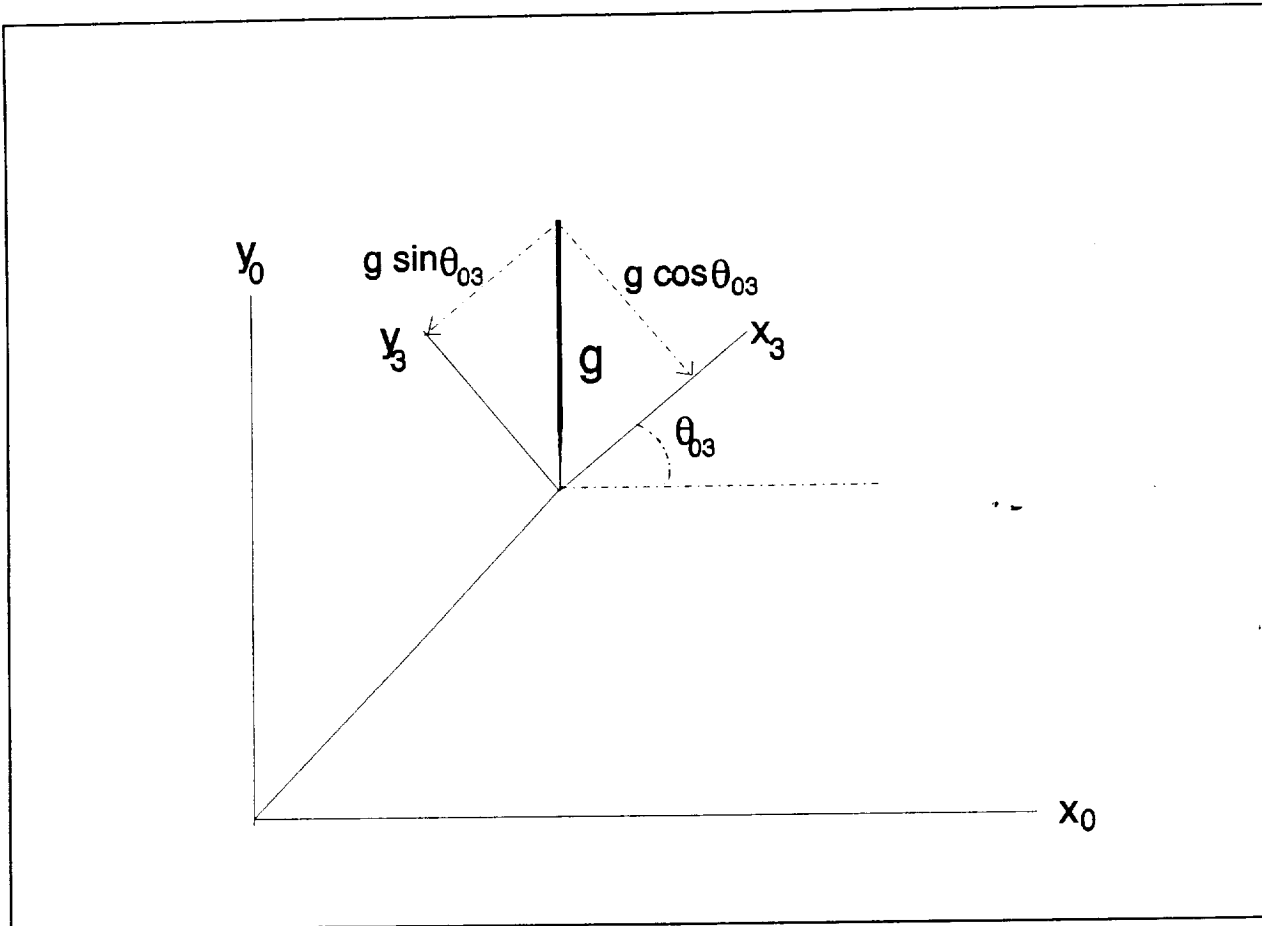


Figure 3.2

position vector of accelerometer 1 on the calf with respect to Frame 3.

Because all motions are assumed to be on the sagittal plane, Equation 3.2 can be written as follows in two dimensions.

$$- {}^3r_{cIy}\alpha_3 - {}^3r_{cIx}\omega_3^2 = {}^3a_{cIx} - {}^3a_{3x} \quad (3.3)$$

$${}^3r_{cIx}\alpha_3 - {}^3r_{cIy}\omega_3^2 = {}^3a_{cIy} - {}^3a_{3y} \quad (3.4)$$

Equation 3.3 and Equation 3.4 can be expressed in matrix form as Equation 3.5.

$${}^3a_{c1} = {}^3a_3 + \begin{bmatrix} -{}^3r_{c1y} & -{}^3r_{c1x} \\ {}^3r_{c1x} & -{}^3r_{c1y} \end{bmatrix} \begin{bmatrix} \alpha_3 \\ \omega_3^2 \end{bmatrix} \quad (3.5)$$

Similarly, the acceleration of accelerometer 2 attached to the calf, denoted  ${}^3a_{c2}$ , can be expressed as follows:

$$-{}^3r_{c2y}\alpha_3 - {}^3r_{c2x}\omega_3^2 = {}^3a_{c2x} - {}^3a_{3x} \quad (3.6)$$

$${}^3r_{c2x}\alpha_3 - {}^3r_{c2y}\omega_3^2 = {}^3a_{c2y} - {}^3a_{3y} \quad (3.7)$$

Equation 3.6 and Equation 3.7 can also be expressed in matrix form as Equation 3.8.

$${}^3a_{c2} = {}^3a_3 + \begin{bmatrix} -{}^3r_{c2y} & -{}^3r_{c2x} \\ {}^3r_{c2x} & -{}^3r_{c2y} \end{bmatrix} \begin{bmatrix} \alpha_3 \\ \omega_3^2 \end{bmatrix} \quad (3.8)$$

Before evaluating  $\omega_3$  and  $\alpha_3$ , the relation between the accelerometer readings and the true accelerations must be established. As shown in Figure 3.2, suppose that the rotation angle between Frame 3 and the inertial frame is  $\theta_{03}$ . Gravity exerts an acceleration of  $-g\sin(\theta_{03})$  in the direction of the x-axis of Frame 3 and an acceleration of  $-g\cos(\theta_{03})$  in the direction of the y-axis of Frame 3. Therefore, the relation between the reading of accelerometer 1 attached to the calf, denoted  $a_{c1}$ , and the actual acceleration, denoted  ${}^3a_{c1}$ , is as follows:

$${}^3a_{c1x} = a_{c1x} + g\sin(\theta_{03}) \quad (3.9)$$

$${}^3a_{c1y} = a_{c1y} + g\cos(\theta_{03}) \quad (3.10)$$

or in matrix form



$${}^3a_{c1} = a_{c1} + \begin{bmatrix} g\sin(\theta_{03}) \\ g\cos(\theta_{03}) \end{bmatrix} \quad (3.11)$$

Similarly, the relation between the actual acceleration of the point of attachment of accelerometer 2, denoted  ${}^3a_{c2}$ , and the reading of accelerometer 2 attached to the calf  $a_{c2}$  is as follows:

$${}^3a_{c2x} = a_{c2x} + g\sin(\theta_{03}) \quad (3.12)$$

$${}^3a_{c2y} = a_{c2y} + g\cos(\theta_{03}) \quad (3.13)$$

or in matrix form.

$${}^3a_{c2} = a_{c2} + \begin{bmatrix} g\sin(\theta_{03}) \\ g\cos(\theta_{03}) \end{bmatrix} \quad (3.14)$$

Until the rotation angles between the inertial frame and Frame 3 or Frame 4 can be determined, the true accelerations of the accelerometers attached to the calf or thigh sections can not be obtained. To simplify the equations, a subscript g will be added to denote an acceleration to which a corresponding gravity component has been added. For example,  ${}^3a_{2g} = {}^3a_2 + {}^3g$ .

Substituting Equation 3.9 and Equation 3.10 into Equation 3.3 and Equation 3.4; Equation 3.12 and Equation 3.13 into Equation 3.6 and Equation 3.7, respectively. The newly defined equations are:

$$- {}^3r_{c1y}\alpha_3 - {}^3r_{c1x}\omega_3^2 = a_{c1x} + g\sin(\theta_{03}) - {}^3a_{3x} \quad (3.15)$$

$${}^3r_{c1x}\alpha_3 - {}^3r_{c1y}\omega_3^2 = a_{c1y} + g\sin(\theta_{03}) - {}^3a_{3y} \quad (3.16)$$

$$- {}^3r_{c2y}\alpha_3 - {}^3r_{c2x}\omega_3^2 = a_{c2x} + g\sin(\theta_{03}) - {}^3a_{3x} \quad (3.17)$$

$${}^3r_{c2x}\alpha_3 - {}^3r_{c2y}\omega_3^2 = a_{c2y} + g\sin(\theta_{03}) - {}^3a_{3y} \quad (3.18)$$

Subtracting Equation 3.15 from Equation 3.17 and Equation 3.16 from Equation 3.18, respectively, and expressing the results in matrix form,

$$\begin{bmatrix} -({}^3r_{c2y}-{}^3r_{c1y}) & -({}^3r_{c2x}-{}^3r_{c1x}) \\ ({}^3r_{c2x}-{}^3r_{c1x}) & -({}^3r_{c2y}-{}^3r_{c1y}) \end{bmatrix} \begin{bmatrix} \alpha_3 \\ \omega_3^2 \end{bmatrix} = a_{c2} - a_{c1} \quad (3.19)$$

Because  ${}^3r_{c1}$  and  ${}^3r_{c2}$  are known and  $a_{c1}$ ,  $a_{c2}$  are measured by the accelerometers, angular velocity and angular acceleration can be evaluated as follows:

$$\begin{bmatrix} -({}^3r_{c2y}-{}^3r_{c1y}) & -({}^3r_{c2x}-{}^3r_{c1x}) \\ ({}^3r_{c2x}-{}^3r_{c1x}) & -({}^3r_{c2y}-{}^3r_{c1y}) \end{bmatrix}^{-1} (a_{c2} - a_{c1}) = \begin{bmatrix} \alpha_3 \\ \omega_3^2 \end{bmatrix} \quad (3.20)$$

### 3.3.2 Determination of Angular Position

Once  $\alpha_3$  and  $\omega_3$  are evaluated, Equation 3.15 and Equation 3.16 can be rewritten as follows:

$$a_{cIx} - (-{}^3r_{cIy}\alpha_3 - {}^3r_{cIx}\omega_3^2) = {}^3a_{3x} - g\sin(\theta_{03}) = {}^3a_{3gx} \quad (3.21)$$

$$a_{cIy} - ({}^3r_{cIx}\alpha_3 - {}^3r_{cIy}\omega_3^2) = {}^3a_{3y} - g\cos(\theta_{03}) = {}^3a_{3gy} \quad (3.22)$$

All the terms on the left-hand side are known; therefore, the terms on the right-hand side, denoted  ${}^3a_{3g}$  --meaning  ${}^3a_3$  with the added gravity component-- can be determined.

From rigid body kinematics, the relation between the accelerations of the knee joint  ${}^3a_3$  and of the foot joint  ${}^3a_2$  is given by:

$$-{}^3r_{32y}\alpha_3 - {}^3r_{32x}\omega_3^2 + {}^3a_{3x} = {}^3a_{2x} \quad (3.23)$$

$${}^3r_{32x}\alpha_3 - {}^3r_{32y}\omega_3^2 + {}^3a_{3y} = {}^3a_{2y} \quad (3.24)$$

where  ${}^3r_{32}$  stands for the position vector of the foot joint with respect to Frame 3.

Adding  $-g\sin(\theta_{03})$  to both sides of Equation 3.23 and  $-g\cos(\theta_{03})$  to both sides of Equation 3.24, respectively, one obtains,

$$-{}^3r_{32y}\alpha_3 - {}^3r_{32x}\omega_3^2 + {}^3a_{3x} - g\sin(\theta_{03}) = {}^3a_{2x} - g\sin(\theta_{03}) \quad (3.25)$$

$${}^3r_{32x}\alpha_3 - {}^3r_{32y}\omega_3^2 + {}^3a_{3y} - g\cos(\theta_{03}) = {}^3a_{2y} - g\cos(\theta_{03}) \quad (3.26)$$

If  ${}^3a_2$  plus its respective gravity component is denoted as  ${}^3a_{2g}$ , the above two equations become:

$$- {}^3r_{32y}\alpha_3 - {}^3r_{32x}\omega_3^2 + {}^3a_{3gx} = {}^3a_{2gx} \quad (3.27)$$

$${}^3r_{32x}\alpha_3 - {}^3r_{32y}\omega_3^2 + {}^3a_{3gy} = {}^3a_{2gy} \quad (3.28)$$

Since the terms on the left-hand side are all known, the ones on the right-hand side,  ${}^3\mathbf{a}_g$ , can be determined.

From robot kinematics theory (Phillip John McKerrow, 1991), the relation between the acceleration of the foot joint with respect to the inertial frame and the acceleration with respect to Frame 3 is as follows:

$$\begin{bmatrix} {}^0a_{2x} \\ {}^0a_{2y} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{03}) & -\sin(\theta_{03}) \\ \sin(\theta_{03}) & \cos(\theta_{03}) \end{bmatrix} \begin{bmatrix} {}^3a_{2x} \\ {}^3a_{2y} \end{bmatrix} \quad (3.29)$$

If  ${}^3\mathbf{a}_2$  is expressed in terms of  ${}^3\mathbf{a}_g$ , then Equation 3.29 becomes Equation 3.30.

$$\begin{bmatrix} {}^0a_{2x} \\ {}^0a_{2y} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{03}) & -\sin(\theta_{03}) \\ \sin(\theta_{03}) & \cos(\theta_{03}) \end{bmatrix} \begin{bmatrix} {}^3a_{2gx} + g\sin(\theta_{03}) \\ {}^3a_{2gy} + g\cos(\theta_{03}) \end{bmatrix} \quad (3.30)$$

In Equation 3.30, the only unknown variable is  $\theta_{03}$ , and it can be expanded as follows:

$${}^0a_{2x} = {}^3a_{2gx}\cos(\theta_{03}) + g\cos(\theta_{03})\sin(\theta_{03}) - {}^3a_{2gy}\sin(\theta_{03}) - g\cos(\theta_{03})\sin(\theta_{03}) \quad (3.31)$$

$${}^0a_{2y} = {}^3a_{2gx}\sin(\theta_{03}) + g\sin^2(\theta_{03}) + {}^3a_{2gy}\cos(\theta_{03}) + g\cos^2(\theta_{03}) \quad (3.32)$$

Then the two equations become:

$${}^0a_{2x} = {}^3a_{2gx}\cos(\theta_{03}) - {}^3a_{2gy}\sin(\theta_{03}) \quad (3.33)$$

$${}^0a_{2y} = {}^3a_{2gx}\sin(\theta_{03}) + {}^3a_{2gy}\cos(\theta_{03}) + g \quad (3.34)$$

Finally  $\theta_{03}$  can be evaluated as:

$$\theta_{03} = \cos^{-1} \left[ \frac{{}^0a_{2x} {}^3a_{2gx} + {}^0a_{2y} {}^3a_{2gy} - g {}^3a_{2gy}}{({}^3a_{2gx})^2 + ({}^3a_{2gy})^2} \right] \quad (3.35)$$

### 3.3.3 Determination of Linear Acceleration

The true acceleration of the knee joint with respect to the inertial frame can thus be evaluated in two steps. First evaluate the true acceleration with respect to Frame 3:

$${}^3a_{3x} = {}^3a_{3gx} + g\sin(\theta_{03}) \quad (3.36)$$

$${}^3a_{3y} = {}^3a_{3gy} + g\cos(\theta_{03}) \quad (3.37)$$

Then the true acceleration of the knee joint with respect to the inertial frame, denoted by  $\mathbf{a}_{03}$ , is obtained as follows:

$$\begin{bmatrix} {}^0a_{3x} \\ {}^0a_{3y} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{03}) & -\sin(\theta_{03}) \\ \sin(\theta_{03}) & \cos(\theta_{03}) \end{bmatrix} \begin{bmatrix} {}^3a_{3x} \\ {}^3a_{3y} \end{bmatrix} \quad (3.38)$$

After finding  $\mathbf{a}_{03}$ , the same procedure is applied to determine the kinematic parameters of the thigh section using the acceleration of the knee joint.

### 3.4 ERROR ANALYSIS

#### 3.4.1 Analysis of error propagation

Accelerometers produce errors in precision and accuracy. If the accuracy error is not negligible, then the accelerometer must be recalibrated. It is assumed that errors are due to precision alone, so that the resultant error of angular position of the moving frame attached to the body member is caused by the propagation of the precision error of the accelerometers. An analysis of accuracy error propagation can be inferred from the precision error analysis and will not be considered in this thesis.

The uncertainty of the measured accelerations is denoted by  $\delta$ . The error analysis follows the procedures used to determine  $\theta_{03}$  and  $\theta_{04}$ .

The only error carrying inputs to determine  $\theta_{03}$  in Equation 3.39 are  ${}^3\mathbf{a}_{2g}$  and  ${}^0\mathbf{a}_2$ . The latter is obtained through calculations using encoder values. The encoders (M20051221031, Dynapar Corp., Gurnee, IL) are of good resolution (2048/rev). It will be assumed that the only sources of error of  $\theta_{03}$  are the values from the accelerometers,  $\mathbf{a}_{c1x}$ ,  $\mathbf{a}_{c1y}$ ,  $\mathbf{a}_{c2x}$  and  $\mathbf{a}_{c2y}$ .

$$\theta_{03} = \cos^{-1} \left[ \frac{{}^0a_{2x} {}^3a_{2gx} + {}^0a_{2y} {}^3a_{2gy} - g {}^3a_{2gy}}{({}^3a_{2gx})^2 + ({}^3a_{2gy})^2} \right] \quad (3.39)$$

Define  $k$  as follows:

$$k = \frac{{}^0a_{2x} {}^3a_{2gx} + {}^0a_{2y} {}^3a_{2gy} - g {}^3a_{2gy}}{({}^3a_{2gx})^2 + ({}^3a_{2gy})^2} \quad (3.40)$$

where  ${}^3\mathbf{a}_{2g}$  is a function of  $\alpha_3$ ,  $\omega_3^2$  and  ${}^3\mathbf{a}_3$  as follows,

$${}^3a_{2gx} = - {}^3r_{32y}\alpha_3 - {}^3r_{32x}\omega_3^2 + {}^3a_{3gx} \quad (3.41)$$

$${}^3a_{2gy} = {}^3r_{32x}\alpha_3 - {}^3r_{32y}\omega_3^2 + {}^3a_{3gy} \quad (3.42)$$

where  ${}^3a_3$  is also a function of  $\mathbf{a}_{c1}$ ,  $\alpha_3$  and  $\omega_3^2$ ,

$${}^3a_{3gx} = a_{c1x} - ( - {}^3r_{c1y}\alpha_3 - {}^3r_{c1x}\omega_3^2 ) \quad (3.43)$$

$${}^3a_{3gy} = a_{c1y} - ( {}^3r_{c1x}\alpha_3 - {}^3r_{c1y}\omega_3^2 ) \quad (3.44)$$

and  $\alpha_3$  and  $\omega_3^2$  are both functions of  $\mathbf{a}_{c1x}$ ,  $\mathbf{a}_{c1y}$ ,  $\mathbf{a}_{c2x}$  and  $\mathbf{a}_{c2y}$ .

$$\alpha_3 = \frac{ - {}^3r_{c2ly}(a_{c2x} - a_{c1x}) + {}^3r_{c2lx}(a_{c2y} - a_{c1y}) }{ {}^3r_{c2lx}^2 + {}^3r_{c2ly}^2 } \quad (3.45)$$

$$\omega_3^2 = \frac{ - {}^3r_{c2lx}(a_{c2x} - a_{c1x}) - {}^3r_{c2ly}(a_{c2y} - a_{c1y}) }{ {}^3r_{c2lx}^2 + {}^3r_{c2ly}^2 } \quad (3.46)$$

Thus  ${}^3a_{2gx}$  and  ${}^3a_{2gy}$  can be expressed in terms of  $\mathbf{a}_{c1x}$ ,  $\mathbf{a}_{c1y}$ ,  $\mathbf{a}_{c2x}$  and  $\mathbf{a}_{c2y}$  as follows.

$$\begin{aligned} {}^3a_{2gx} = & \{ [ ( {}^3r_{c21x}^2 + {}^3r_{c21y}^2 ) + {}^3r_{c21y}({}^3r_{c1y} - {}^3r_{32y}) + {}^3r_{c21x}({}^3r_{c1x} - {}^3r_{32x}) ] \mathbf{a}_{c1x} + \\ & [ {}^3r_{c21y}({}^3r_{c1x} - {}^3r_{32x}) - {}^3r_{c21x}({}^3r_{c1y} - {}^3r_{32y}) ] \mathbf{a}_{c1y} + \\ & [ - {}^3r_{c21y}({}^3r_{c1y} - {}^3r_{32y}) - {}^3r_{c21x}({}^3r_{c1x} - {}^3r_{32x}) ] \mathbf{a}_{c2x} + \\ & [ {}^3r_{c21x}({}^3r_{c1y} - {}^3r_{32y}) - {}^3r_{c21y}({}^3r_{c1x} - {}^3r_{32x}) ] \mathbf{a}_{c2y} \\ & \} / [ {}^3r_{c21x}^2 + {}^3r_{c21y}^2 ] \end{aligned} \quad (3.47)$$

$$\begin{aligned}
{}^3\mathbf{a}_{2gy} = & \{ [- {}^3\mathbf{r}_{c2ly}({}^3\mathbf{r}_{c1x}-{}^3\mathbf{r}_{32x}) + {}^3\mathbf{r}_{c21x}({}^3\mathbf{r}_{c1y}-{}^3\mathbf{r}_{32y}) ] \mathbf{a}_{c1x} + \\
& [ ({}^3\mathbf{r}_{c21x}^2 + {}^3\mathbf{r}_{c21y}^2) + {}^3\mathbf{r}_{c21y}({}^3\mathbf{r}_{c1y}-{}^3\mathbf{r}_{32y}) + {}^3\mathbf{r}_{c21x}({}^3\mathbf{r}_{c1x}-{}^3\mathbf{r}_{32x}) ] \mathbf{a}_{c1y} + \\
& [- {}^3\mathbf{r}_{c21x}({}^3\mathbf{r}_{c1y}-{}^3\mathbf{r}_{32y}) + {}^3\mathbf{r}_{c21y}({}^3\mathbf{r}_{c1x}-{}^3\mathbf{r}_{32x}) ] \mathbf{a}_{c2x} + \\
& [- {}^3\mathbf{r}_{c21y}({}^3\mathbf{r}_{c1y}-{}^3\mathbf{r}_{32y}) - {}^3\mathbf{r}_{c21x}({}^3\mathbf{r}_{c1x}-{}^3\mathbf{r}_{32x}) ] \mathbf{a}_{c2y} \\
& \} / [{}^3\mathbf{r}_{c21x}^2 + {}^3\mathbf{r}_{c21y}^2] \quad (3.48)
\end{aligned}$$

where  ${}^3\mathbf{r}_{c21} = {}^3\mathbf{r}_{c2} - {}^3\mathbf{r}_{c1}$  is the vector from accelerometer 1 to accelerometer 2 expressed in Frame 3.

$\theta_{03}$  is defined by:

$$\theta_{03} = \cos^{-1}(k) = \cos^{-1}[k({}^3a_{2gx}, {}^3a_{2gy})] \quad (3.49)$$

which may be expressed as follows:

$$\theta_{03} = \cos^{-1}(k[{}^3a_{2gx}(a_{c1x}, a_{c1y}, a_{c2x}, a_{c2y}), {}^3a_{2gy}(a_{c1x}, a_{c1y}, a_{c2x}, a_{c2y})]) \quad (3.50)$$

The square of the uncertainty of  $\theta_{03}$  is defined by:

$$S_{\theta_{03}}^2 = \left(\frac{\partial \theta_{03}}{\partial a_{c1x}}\right)^2 (\delta a_{c1x})^2 + \left(\frac{\partial \theta_{03}}{\partial a_{c1y}}\right)^2 (\delta a_{c1y})^2 + \left(\frac{\partial \theta_{03}}{\partial a_{c2x}}\right)^2 (\delta a_{c2x})^2 + \left(\frac{\partial \theta_{03}}{\partial a_{c2y}}\right)^2 (\delta a_{c2y})^2 \quad (3.51)$$

Using the chain rule, the partial derivative terms of the preceding equation are determined.

$$\frac{\partial \theta_{03}}{\partial a_{c1x}} = \frac{\partial \theta_{03}}{\partial k} \frac{\partial k}{\partial {}^3a_{2gx}} \frac{\partial {}^3a_{2gx}}{\partial a_{c1x}} + \frac{\partial \theta_{03}}{\partial k} \frac{\partial k}{\partial {}^3a_{2gy}} \frac{\partial {}^3a_{2gy}}{\partial a_{c1x}} \quad (3.52)$$



$$\frac{\partial \theta_{03}}{\partial a_{c1y}} = \frac{\partial \theta_{03}}{\partial k} \frac{\partial k}{\partial {}^3a_{2gx}} \frac{\partial {}^3a_{2gx}}{\partial a_{c1y}} + \frac{\partial \theta_{03}}{\partial k} \frac{\partial k}{\partial {}^3a_{2gy}} \frac{\partial {}^3a_{2gy}}{\partial a_{c1y}} \quad (3.53)$$

$$\frac{\partial \theta_{03}}{\partial a_{c2x}} = \frac{\partial \theta_{03}}{\partial k} \frac{\partial k}{\partial {}^3a_{2gx}} \frac{\partial {}^3a_{2gx}}{\partial a_{c2x}} + \frac{\partial \theta_{03}}{\partial k} \frac{\partial k}{\partial {}^3a_{2gy}} \frac{\partial {}^3a_{2gy}}{\partial a_{c2x}} \quad (3.54)$$

$$\frac{\partial \theta_{03}}{\partial a_{c2y}} = \frac{\partial \theta_{03}}{\partial k} \frac{\partial k}{\partial {}^3a_{2gx}} \frac{\partial {}^3a_{2gx}}{\partial a_{c2y}} + \frac{\partial \theta_{03}}{\partial k} \frac{\partial k}{\partial {}^3a_{2gy}} \frac{\partial {}^3a_{2gy}}{\partial a_{c2y}} \quad (3.55)$$

where

$$\frac{\partial \theta_{03}}{\partial k} = -\frac{1}{\sqrt{1-k^2}} \quad (3.56)$$

$$\frac{\partial k}{\partial {}^3a_{2gx}} = \frac{{}^0a_{2x}({}^3a_{2gx}^2 + {}^3a_{2gy}^2) - 2 {}^3a_{2gx}({}^0a_{2x} {}^3a_{2gx} + {}^0a_{2y} {}^3a_{2gy} - g {}^3a_{2gy})}{({}^3a_{2gx}^2 + {}^3a_{2gy}^2)^2} \quad (3.57)$$

$$\frac{\partial k}{\partial {}^3a_{2gy}} = \frac{({}^0a_{2y} - g)({}^3a_{2gx}^2 + {}^3a_{2gy}^2) - 2 {}^3a_{2gy}({}^0a_{2x} {}^3a_{2gx} + {}^0a_{2y} {}^3a_{2gy} - g {}^3a_{2gy})}{({}^3a_{2gx}^2 + {}^3a_{2gy}^2)^2} \quad (3.58)$$

$$\frac{\partial {}^3a_{2gx}}{\partial a_{c1x}} = \frac{({}^3r_{c21x}^2 + {}^3r_{c21y}^2) + {}^3r_{c21y}({}^3r_{c1y} - {}^3r_{32y}) + {}^3r_{c21x}({}^3r_{c1x} - {}^3r_{32x})}{{}^3r_{c21x}^2 + {}^3r_{c21y}^2} \quad (3.59)$$

$$\frac{\partial {}^3a_{2gx}}{\partial a_{c1y}} = \frac{{}^3r_{c21y}({}^3r_{c1x} - {}^3r_{32x}) - {}^3r_{c21x}({}^3r_{c1y} - {}^3r_{32y})}{{}^3r_{c21x}^2 + {}^3r_{c21y}^2} \quad (3.60)$$

$$\frac{\partial {}^3a_{2gx}}{a_{c2x}} = \frac{{}^3r_{c21y}({}^3r_{c1y} - {}^3r_{32y}) - {}^3r_{c21x}({}^3r_{c1x} - {}^3r_{32x})}{{}^3r_{c21x}^2 + {}^3r_{c21y}^2} \quad (3.61)$$

$$\frac{\partial {}^3a_{2gx}}{\partial a_{c2y}} = \frac{{}^3r_{c21x}({}^3r_{c1y} - {}^3r_{32y}) - {}^3r_{c21y}({}^3r_{c1x} - {}^3r_{32x})}{{}^3r_{c21x}^2 + {}^3r_{c21y}^2} \quad (3.62)$$

$$\frac{\partial {}^3a_{2gy}}{\partial a_{c1x}} = \frac{-{}^3r_{c21y}({}^3r_{c1x} - {}^3r_{32x}) + {}^3r_{c21x}({}^3r_{c1y} - {}^3r_{32y})}{{}^3r_{c21x}^2 + {}^3r_{c21y}^2} \quad (3.63)$$

$$\frac{\partial {}^3a_{2gy}}{\partial a_{c1y}} = \frac{({}^3r_{c21x}^2 + {}^3r_{c21y}^2) + {}^3r_{c21y}({}^3r_{c1y} - {}^3r_{32y}) + {}^3r_{c21x}({}^3r_{c1x} - {}^3r_{32x})}{{}^3r_{c21x}^2 + {}^3r_{c21y}^2} \quad (3.64)$$

$$\frac{\partial {}^3a_{2gy}}{\partial a_{c2x}} = \frac{- {}^3r_{c2lx}({}^3r_{c1y} - {}^3r_{32y}) + {}^3r_{c2ly}({}^3r_{c1x} - {}^3r_{32x})}{{}^3r_{c2lx}^2 + {}^3r_{c2ly}^2} \quad (3.65)$$

$$\frac{\partial {}^3a_{2gy}}{a_{c2y}} = \frac{- {}^3r_{c2ly}({}^3r_{c1y} - {}^3r_{32y}) - {}^3r_{c2lx}({}^3r_{c1x} - {}^3r_{32x})}{{}^3r_{c2lx}^2 + {}^3r_{c2ly}^2} \quad (3.66)$$

In Equation 3.56, when  $k$  approaches one, the partial derivative of  $\theta_{03}$  approaches negative infinity. This presents a practical problem to the error analysis, so the error of  $k$  is analyzed instead of the error of  $\theta_{03}$ .

The square of uncertainty of the  $k$  is as follow:

$$S_k^2 = \left(\frac{\partial k}{\partial a_{c1x}}\right)^2 (\delta a_{c1x})^2 + \left(\frac{\partial k}{\partial a_{c1y}}\right)^2 (\delta a_{c1y})^2 + \left(\frac{\partial k}{\partial a_{c2x}}\right)^2 (\delta a_{c2x})^2 + \left(\frac{\partial k}{\partial a_{c2y}}\right)^2 (\delta a_{c2y})^2 \quad (3.67)$$

Using the chain rule, the partial derivatives terms in Equation 3.67 are defined as follows.

$$\frac{\partial k}{\partial a_{c1x}} = \frac{\partial k}{\partial {}^3a_{2gx}} \frac{\partial {}^3a_{2gx}}{\partial a_{c1x}} + \frac{\partial k}{\partial {}^3a_{2gy}} \frac{\partial {}^3a_{2gy}}{\partial a_{c1x}} \quad (3.68)$$

$$\frac{\partial k}{\partial a_{c1y}} = \frac{\partial k}{\partial {}^3a_{2gx}} \frac{\partial {}^3a_{2gx}}{\partial a_{c1y}} + \frac{\partial k}{\partial {}^3a_{2gy}} \frac{\partial {}^3a_{2gy}}{\partial a_{c1y}} \quad (3.69)$$

$$\frac{\partial k}{\partial a_{c2x}} = \frac{\partial k}{\partial {}^3a_{2gx}} \frac{\partial {}^3a_{2gx}}{\partial a_{c2x}} + \frac{\partial k}{\partial {}^3a_{2gy}} \frac{\partial {}^3a_{2gy}}{\partial a_{c2x}} \quad (3.70)$$

$$\frac{\partial k}{\partial a_{c2y}} = \frac{\partial k}{\partial {}^3a_{2gx}} \frac{\partial {}^3a_{2gx}}{\partial a_{c2y}} + \frac{\partial k}{\partial {}^3a_{2gy}} \frac{\partial {}^3a_{2gy}}{\partial a_{c2y}} \quad (3.71)$$

The partial derivative terms of Equations 3.68-3.71 have been determined as Equations 3.56-3.66.

The square of uncertainty of  $k$  is thus determined using Equations 3.67-3.71. Finally, the square of the uncertainties of  $\theta_{03}$  can also be determined using the equation below.

$$S_{\theta_{03}}^2 = S_k^2 \left( \frac{\partial \theta_{03}}{\partial k} \right)^2 \quad (3.72)$$

In Equation 3.72, the value of the partial derivative term affects the outcome of the square of uncertainty in  $\theta_{03}$ . The partial derivative term has been defined as Equation 3.56. Within 90 degrees, it is seen that since  $\cos(\theta_{03})$  is decreasing, the square of the partial derivative term is also decreasing. This means that less  $\theta_{03}$  results in larger value of the square of the partial derivative term, and thus in larger error. Therefore,  $\cos(\theta_{03})$  is not suitable for error analysis 90 degrees, and  $\sin(\theta_{03})$  should be used. Conversely,  $\cos(\theta_{03})$  is more suitable than  $\sin(\theta_{03})$  within 90 up to 180 degrees.

From Equations 3.53 and 3.54,  $\theta_{03}$  can also be evaluated as:

$$\theta_{03} = \sin^{-1} \left[ \frac{({}^0a_{2y} - g)^3 a_{2gx} - {}^0a_{2x}^3 a_{2gy}}{({}^3a_{2gx})^2 + ({}^3a_{2gy})^2} \right] \quad (3.73)$$

Also,  $\mathbf{k}$  should be defined as  $\sin(\theta_{03})$  and thus Equations 3.57 and 3.58 become:

$$\frac{\partial k}{\partial {}^3a_{2gx}} = \frac{({}^0a_{2y} - g)({}^3a_{2gx}^2 + {}^3a_{2gy}^2) - 2 {}^3a_{2gx}({}^0a_{2y} {}^3a_{2gx} - {}^0a_{2x} {}^3a_{2gy} - g {}^3a_{2gx})}{({}^3a_{2gx}^2 + {}^3a_{2gy}^2)^2} \quad (3.74)$$

$$\frac{\partial k}{\partial {}^3a_{2gy}} = \frac{- {}^0a_{2x}({}^3a_{2gx}^2 + {}^3a_{2gy}^2) - 2 {}^3a_{2gy}({}^0a_{2y} {}^3a_{2gx} - {}^0a_{2x} {}^3a_{2gy} - g {}^3a_{2gx})}{({}^3a_{2gx}^2 + {}^3a_{2gy}^2)^2} \quad (3.75)$$

Yet Equations 3.59-3.66 remain the same.

### 3.4.2 Numerical demonstration

Based on the analysis in the previous section, a numerical estimation of the error of  $\mathbf{k}$  was performed. The numerical data sets used as inputs are designed based on assumptions that angular velocities of all the sections are constant and angular accelerations are all zero. Other assumptions to simplify the creation of the numerical data are that the hip remains static, the rider-bicycle system is modeled as a five-bar linkage, and the pedal positions are always horizontal. The five-bar linkage system and position vectors on the calf section are shown in Figure 3.3. If the angular position of the crank is given, the angular positions of the other moving sections can be determined. At the same time, the angular velocities of the calf and thigh are chosen to be respectively one fourth and one sixth of the angular velocity of the crank chosen to be 6 rad/sec. Thus the angular velocities of the calf and thigh are respectively 1.5 and 1 rad/sec. Other numerical data are defined as follows: accelerometer precision errors are assumed to be 1%, 3%, 5% and 10%; and  ${}^3\mathbf{r}_{c1x} = -0.10$  m,  ${}^3\mathbf{r}_{c1y} = -0.05$  m,  ${}^3\mathbf{r}_{c21y} = 0$  m,

${}^3\mathbf{r}_{32x} = -0.40 \text{ m}$ ,  ${}^3\mathbf{r}_{32y} = 0 \text{ m}$  and  ${}^3\mathbf{r}_{c21y} = 0 \text{ m}$ . To see the effects of both the distance between the two accelerometers and the precision errors of accelerometer,  ${}^3\mathbf{r}_{c21x}$  is chosen to be  $-0.20 \text{ m}$  and  $-0.3 \text{ m}$ . Using the preceding position vectors, Equations 3.59-3.66 can be determined.

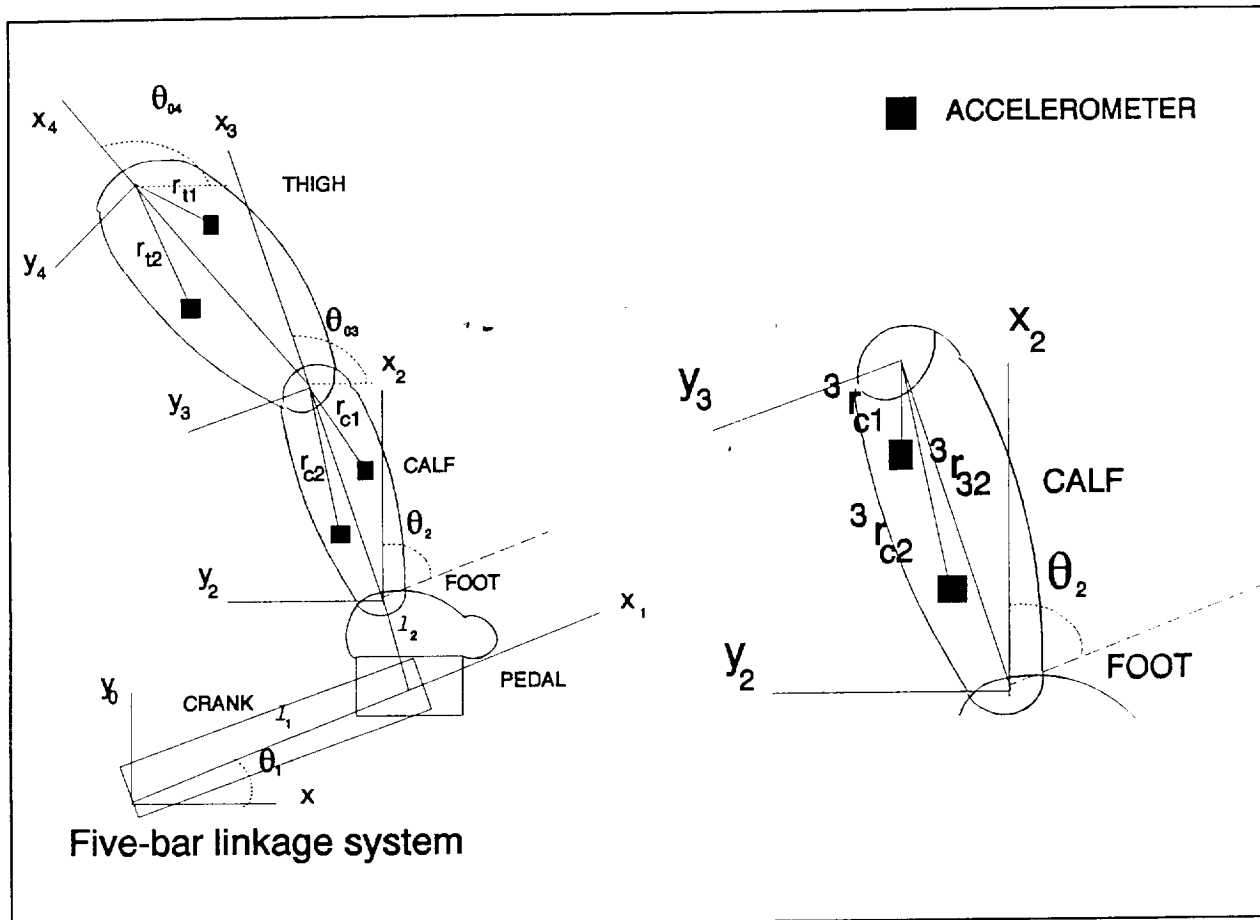


Figure 3.3

In this numerical demonstration, two MATLAB programs TEST.M and KINET.M are used and their descriptions are respectively in Sections 4.3 and 4.4. The TEST.M program produces accelerometer values for simulating the kinematics model. The KINET.M determines the kinematic parameters which have been described in Section 3.3, Kinematic equations. From the geometry of the five-bar linkage model,  $\theta_1$  is given and  $\theta_2$  is determined from the condition

that the foot remains horizontal, thus  $\theta_{03}$  and  $\theta_{04}$  can be determined. Because  $\theta_{03}$  in this five-bar linkage model is less than 90 degrees,  $k$  is defined as  $\sin(\theta_{03})$ . The accelerometer values are produced by TEST.M using the above  $\theta_1, \theta_2, \theta_{03}, \theta_{04}$  and the designed angular velocities of the moving sections. Then KINET.M determines  ${}^3\mathbf{a}_{2g}$  and  ${}^0\mathbf{a}_2$ . Once  ${}^3\mathbf{a}_{2g}$  and  ${}^0\mathbf{a}_2$  are known, Equation 3.57 and 3.58 can be evaluated. Finally, the uncertainty of  $k$  is obtained using the variations of the accelerometer values (Equation 3.76). The resultant variations of  $\theta_{03}$  can be evaluated using Equation 3.72 and the results are shown in Figures 3.4-3.5.

These two figures show error calculations at different distances between accelerometers. Each of the four curves in each figure represents different accelerometer precision errors of 1%, 3%, 5% and 10%. The results shown in Figures 3.4 and 3.5 are at the same crank's angular velocity--6 rad/sec, but at different distances between accelerometers.

From Equations 3.59-3.66, it can be seen that increasing the separation between accelerometers decreases the error. This is proven by comparing Figure 3.4 and 3.5. The results show that within acceptable accelerometer precision errors (<5%), the expected error is probably within the specifications of practical applications.

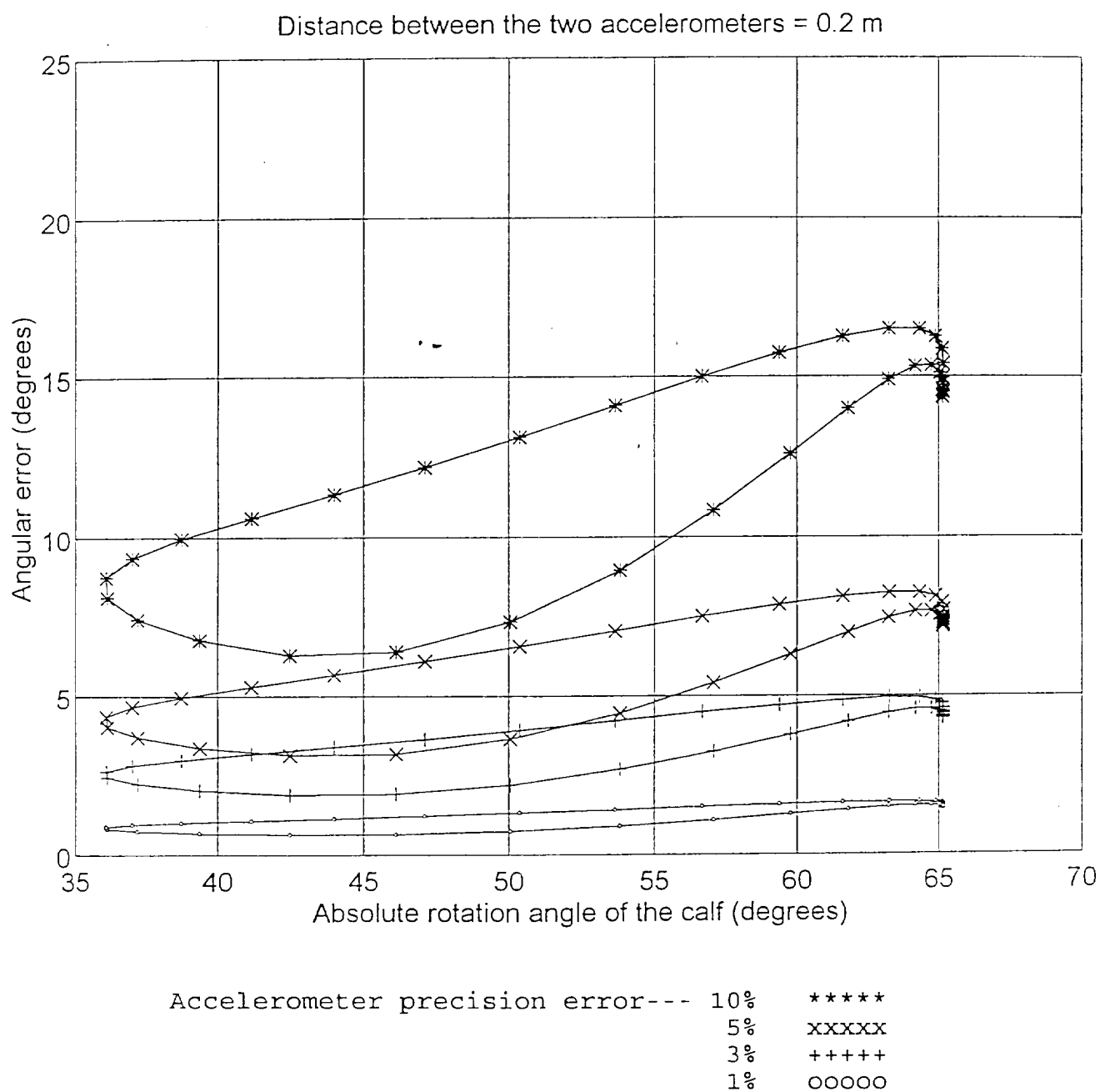


Figure 3.4

Dependency of the error of the calf angle on % accelerometer





## **SOFTWARE**

### **4.1 INTRODUCTION**

The task achieved by the data acquisition program in C++ is to record synchronously the data from accelerometers and encoders, then to evaluate the angular positions, angular velocities and angular accelerations of the crank and pedals during exercise. A second program in MATLAB uses the data from the C++ program to determine the angular orientations, angular velocities, angular accelerations of the calf and thigh sections, and the linear accelerations of the knee joint and the hip joint.

### **4.2 THE DATA ACQUISITION PROGRAM IN C++**

During the development of the software for data acquisition, five programs were created: SCANOPF.CPP, SCANOPF2.CPP, SCANOPB.CPP, SCANOPB2.CPP and SCANSTAR.CPP. Only SCANOPB.CPP determines angular positions, angular velocities and angular accelerations for the kinematics program in MATLAB. Each program served a purpose within the research. Figures 4.2--4.6 show the flowcharts of the five programs. These programs differ from each other primarily in the way that they acquire and store data. The reason for creating the programs was to find a way to quickly digitize a large number of samples. Table 4.1 shows the differences inside the data acquisition loops. The differences between SCANOPF.CPP, SCANOPB.CPP and SCANSTAR.CPP are in the ways that data is stored. The differences between SCANOPF.CPP and SCANOPF2.CPP, SCANOPB.CPP and

SCANOPB2.CPP are that both SCANOPF2.CPP and SCANOPB2.CPP only acquire data from the A/D board but both SCANOPF.CPP and SCANOPB.CPP acquire data from the A/D board and the decoder board. The difference between the SCANSTAR.CPP and the other four is that the data buffer used in SCANSTAR.CPP is allocated by NI-DAQ but the buffers in the other four programs are allocated by DOS. The differences are explained in more detail later.

#### 4.2.1 Storage of Data

The functions in the NI-DAQ library for scanning the channels are `SCAN_Op` and `SCAN_Start`, and both can store data only into a buffer but not a file. The form of the data in the buffer is binary. This cannot be read directly from the screen, so it must be converted into ASCII text form (NI-DAQ Function Reference Manual).

There are three possible methods to store data. The three methods are used by the five data acquisition programs, their differences are shown in Table 4.1 and their flowcharts appear in Figures 4.2-4.6. The first method, used by SCANOPF.CPP and SCANOPF2.CPP, saves the acquired data from the buffer into a binary file immediately after each scan (i.e., save inside the data acquisition loop). Then the binary file is converted into a text file after the end of the entire data acquisition operation. The reason that the binary data in the buffer is not directly stored into a text file inside the data acquisition loop is to reduce the time inside the loop. The second method, used by SCANOPB.CPP and SCANOPB2.CPP, uses a buffer big enough for the entire data acquisition process, and then saves the data from the buffer into a text file. The difference between the first two methods is the size of the buffer used by the data acquisition operation. The first method requires a buffer only big enough for one scan but the second needs

Table 4.1 Differences inside data acquisition loop

Differences Files	Acquires data from the decoder board	Buffer		A file to store data
		Size	Allocated by	
SCANOPF.CPP	Yes	Only big enough for one scan	DOS	Yes
SCANOPF2.CPP	No			
SCANOPB.CPP	Yes	Big enough to store the whole data of many scans	DOS	No
SCANOPB2.CPP	No			
SCANSTAR.CPP	Yes	Big enough to store the whole data of many scans	NI-DAQ functions	No

a buffer big enough for the entire data set. The third method is used by SCANSTAR.CPP and uses a buffer created by a NI-DAQ function, whose size can exceed 64k.

The second method encounters the fact that Turbo C++ (Version 3.0, Borland International Inc., Scotts Valley, CA) normally limits the size of all static data to 64K. The Huge Model sets aside that limit (Turbo C++ User's Guide). Yet the library functions offered by NI-DAQ are for the Large Model so they cannot work in the Huge Model. The first method avoids the buffersize limitation at the expense of reducing the rate of data acquisition, because it requires extra time in the data acquisition loop to store data into a file. The shortcoming of the third method is that the buffer is not available for synchronous data acquisition operations; thus, only SCAN\_Start can use the third method. Therefore, only SCANOPB.CPP determines angular positions, angular velocities and angular accelerations, which are later used by the program for kinematics in MATLAB.

#### **4.2.2 Synchronization with encoders**

Using the NI-DAQ function `SCAN_Op`, synchronous, multiple-channel scanned data acquisition operations are performed (NI-DAQ Function Reference Manual). Using `SCAN_Op`, synchronization is achieved by an internal on-board counter that produces trigger pulses. Thus A/D conversions do not begin until a trigger pulse is applied. This is controlled by two functions in NI-DAQ--`DAQ_Config` and `CTR_Square`. The former concerns the configuration information for subsequent data acquisition operations (NI-DAQ Function Reference Manual) and the latter programs a counter to generate a continuous square wave output of specified duty cycle and frequency (NI-DAQ Function Reference Manual). Therefore, the rate of trigger pulse of the counter defines the rate of operation of `SCAN_Op`. Each operation of `SCAN_Op` is immediately followed by a reading of the decoder board.

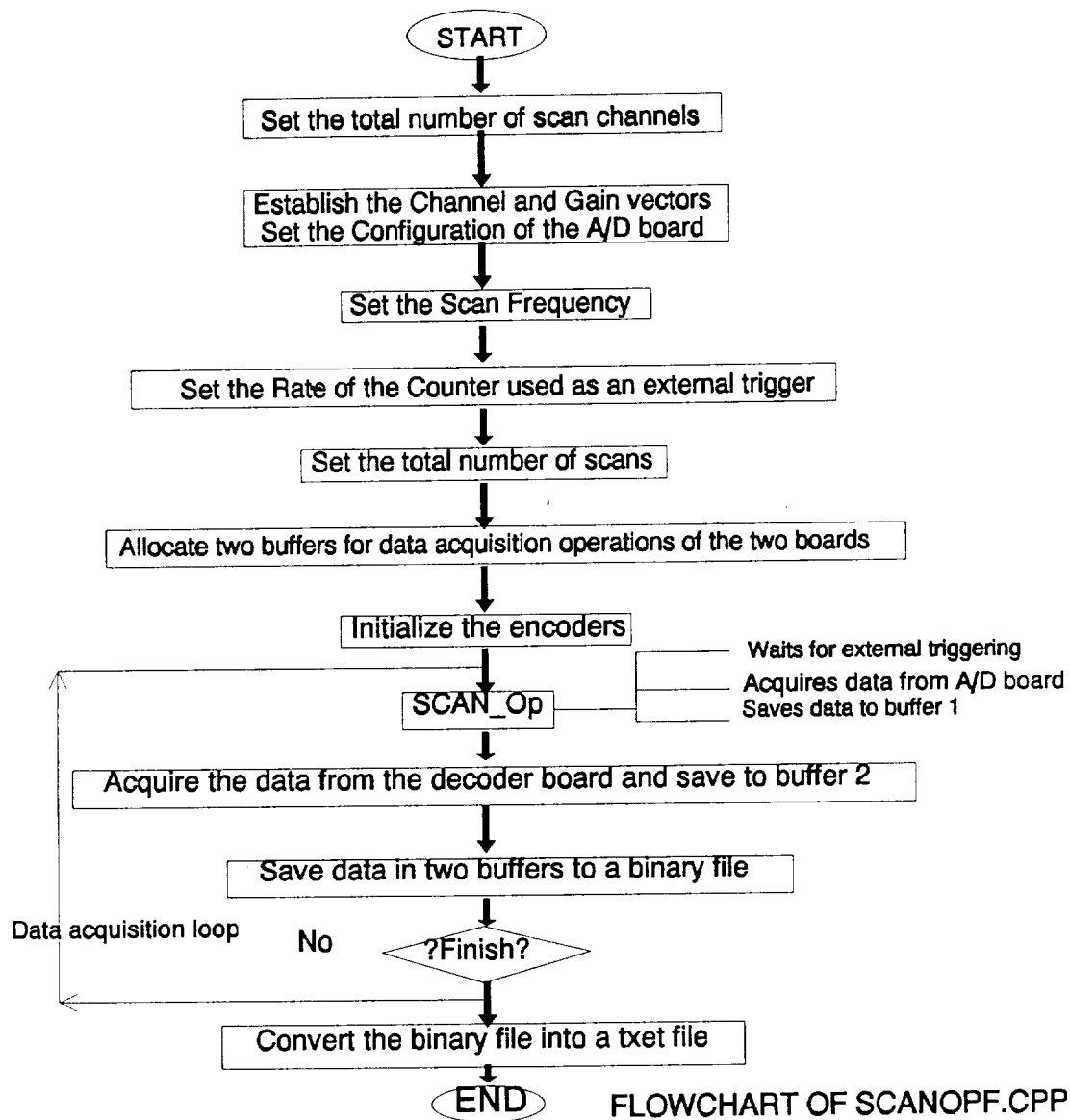


Figure 4.1

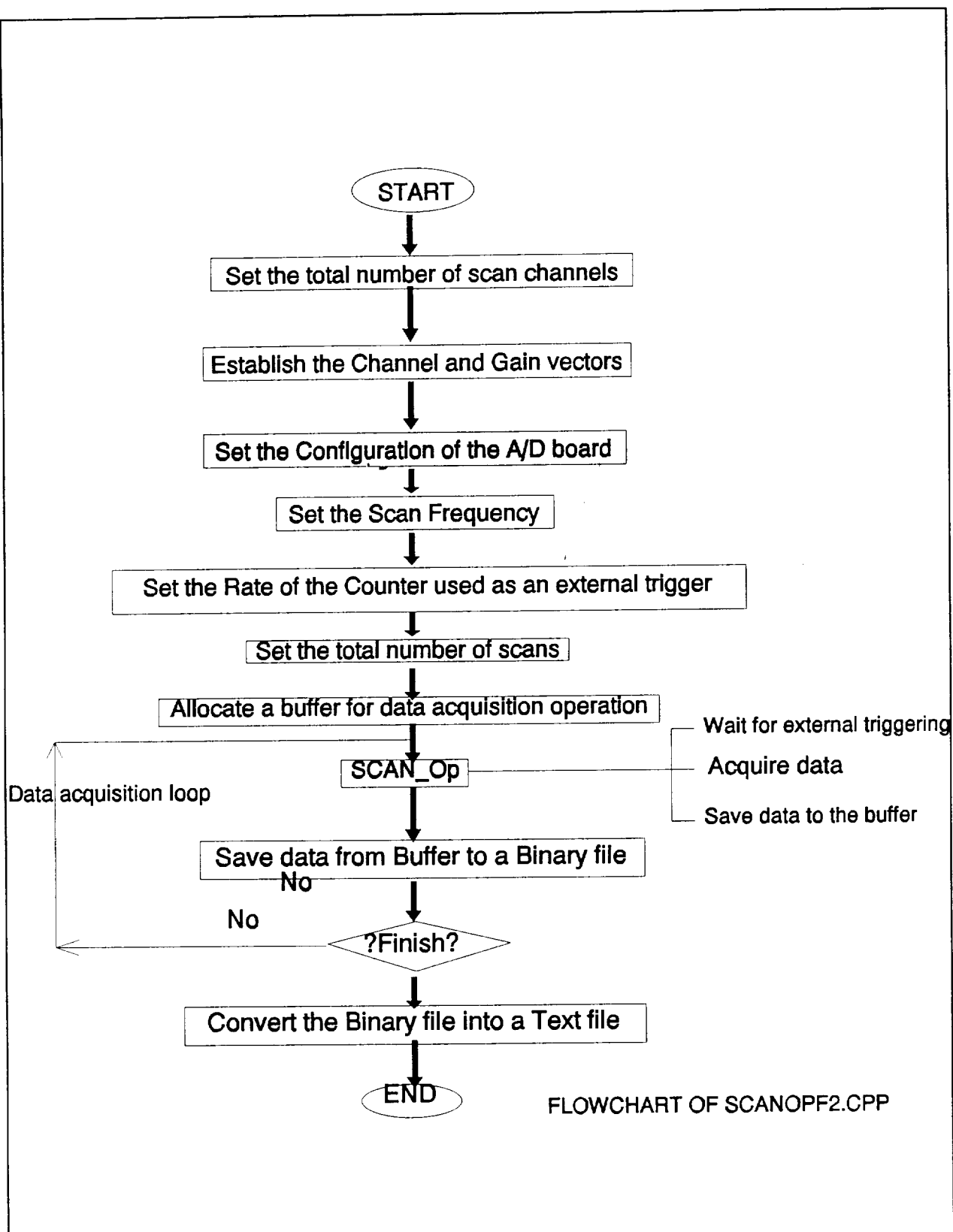


Figure 4.2

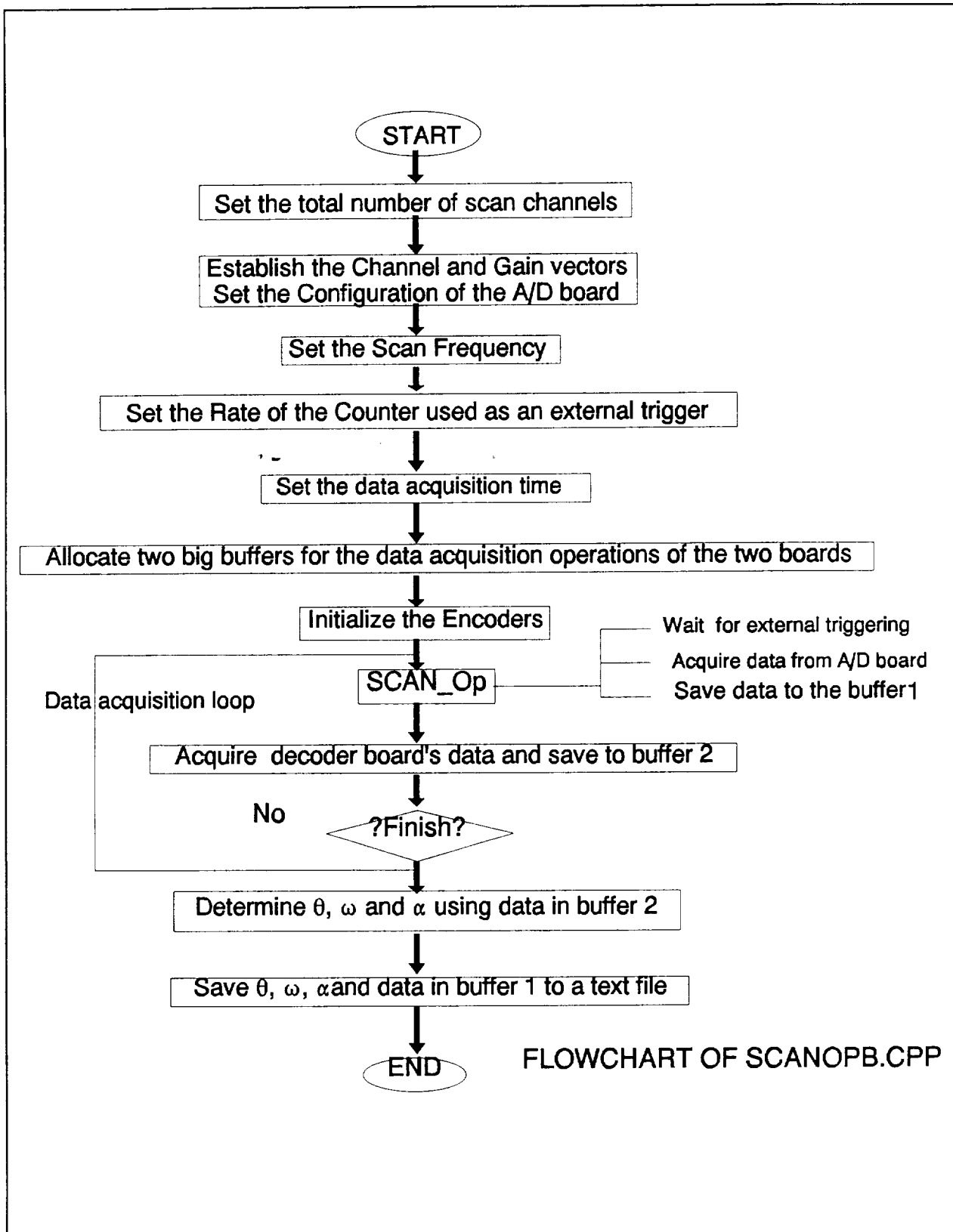


Figure 4.3



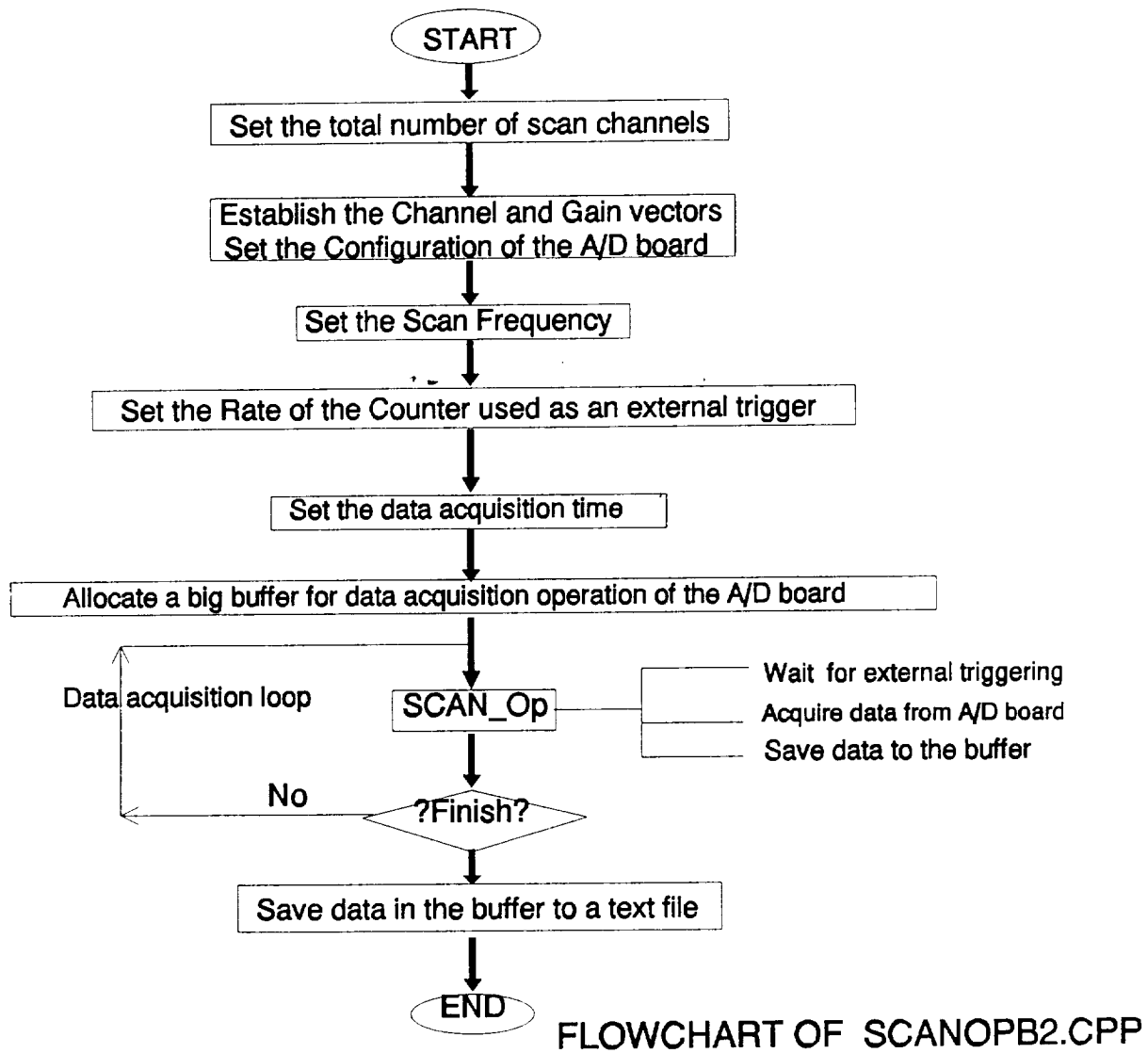


Figure 4.4

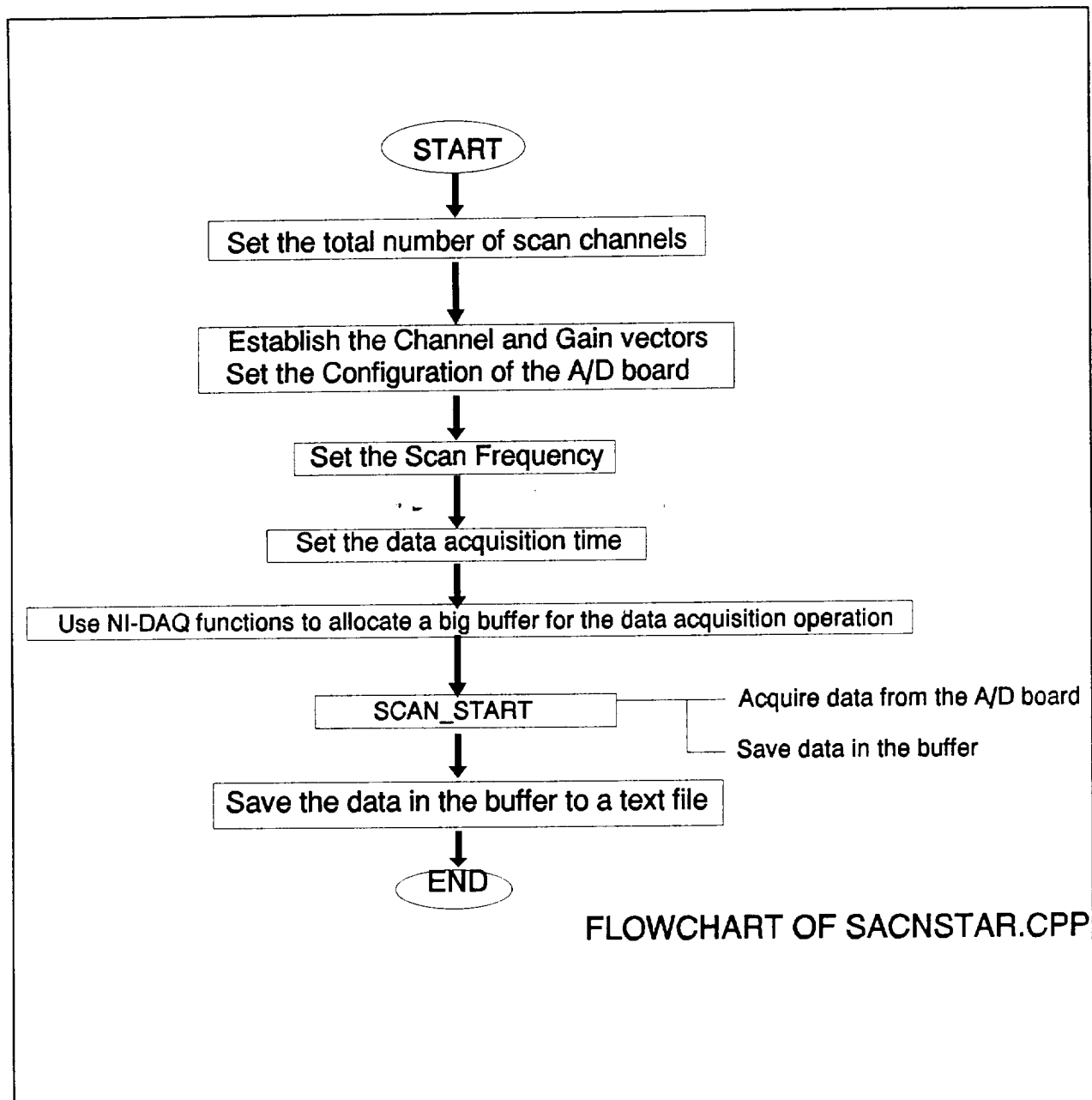


Figure 4.5

### 4.2.3 Checking the D/A System

From the algorithms shown in Figure 4.2 and Figure 4.4, the time difference between the starting of the data acquisition operation of the A/D board and the subsequent waiting of SCAN\_Op for the trigger pulse must be less than the period of the trigger pulse; otherwise, some trigger pulses may be missed. Whether trigger pulses are missed can be checked by inspecting a digitized signal whose original form was known already. If digitization does not affect the integrity of the signal, no trigger pulse is missed and the period is greater than the time difference. For example, given a sinusoidal signal at 5 Hz and the rate of the counter is 50 Hz, the digitized sinusoidal signal should clearly reveal that ten data points are shown in a period if no trigger pulse is missed; otherwise, some trigger pulses are lost. Through this method the maximum rates of the entire data acquisition operation corresponding to different methods of storage were decided. The maximum rate of data acquisition in SCANOPF.CPP is 57Hz, the one of SCANOPB.CPP is 65Hz, and the one of SCANSTAR.CPP is above 1.2 kHz.

### 4.2.4 Discussion

The difference of the maximum rate of data acquisition between SCANSTAR.CPP and the other two is remarkable. It is necessary to analyze the essential difference between them to understand the cause of the big drop of the data acquisition rate.

The main difference in the data acquisition algorithm of SCANSTAR.CPP and the other two is synchronization. SCANSATR.CPP does not read the decoder board. It only reads the DA board and was created as a research tool only. To synchronize the DA board and the decoder

board, SCANOPF.CPP and SCANOPB.CPP use a loop to include the data acquisition operation of the decoder board. Both the loop and the inclusion of the decoder board's data acquisition operation reduce the data acquisition rate. If the main cause of the time delay was from the data acquisition operation of the decoder board, the big drop should be reduced by excluding the decoder board's data acquisition operation from the data acquisition loop. If the exclusion of the decoder board's data acquisition operation cannot reduce the big drop, the cause is in the call to SCAN\_Op.

After the decoder board's data acquisition operation is excluded from the loops of SCANOPF.CPP and SCANOPB.CPP, the two programs become SCANOPF2.CPP and SCANOPB2.CPP, respectively. Figure 4.2 and Figure 4.4 show the flowcharts. The maximum rates of the two programs are 63Hz and 67Hz. This reveals that the main cause of the big drop of the data acquisition is due to the call to SCAN\_Op.

A limiting factor in the system is that the most A/D conversions performed by any data acquisition function in NI-DAQ is 65535. This limit prevents long data acquisition time at high rate. The programs that save data to a file after every scan can acquire data as long as there is space in the hard disk, but the rate may be too slow for some applications.

### 4.3 THE MATLAB PROGRAM FOR KINEMATICS

The program, KINET.M, uses the parameters  $\theta_1, \theta_2, \alpha_1, \alpha_2, \omega_1, \omega_2$  and accelerometer values from an external file made by SCANOPB.CPP to determine the angular positions, angular velocities, angular accelerations and linear accelerations of the calf and thigh sections during pedaling. Its flowchart is shown in Figure 4.7. Section 3.3, Kinematic analysis, has described

the theoretical background of the program.

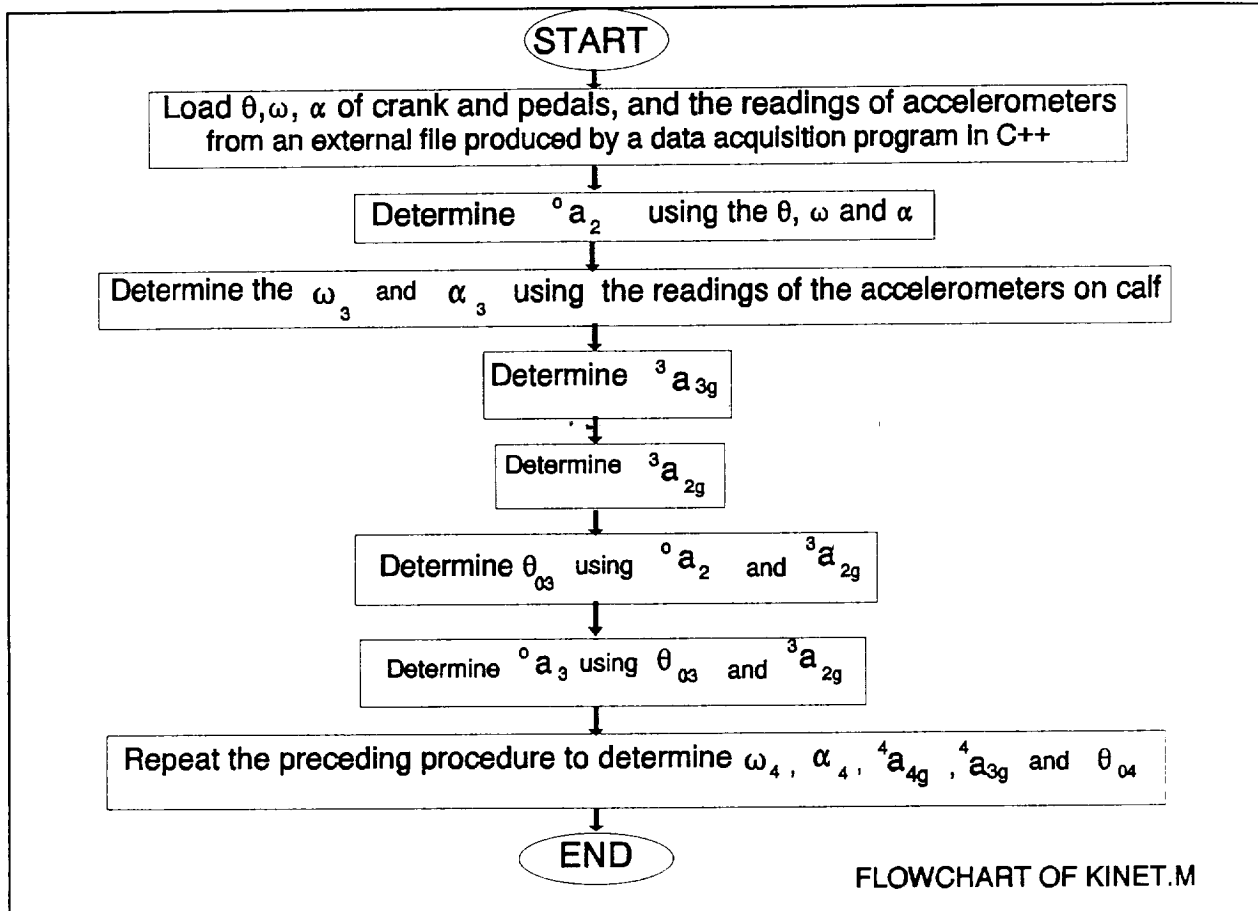


Figure 4.6 The flowchart of KINET.M

#### 4.4 SIMULATION

Using as input  $\theta_1, \theta_2, \alpha_1, \alpha_2, \omega_1, \omega_2$  and the accelerometers' measurements, KINET.M determines  $\alpha_3, \alpha_4, \omega_3, \omega_4$  and the rotation angles between the moving frames and the inertial frame. A simulation was done to make sure that KINET.M works appropriately.

Many sets of test values were used as inputs to a test program, TEST.M. Every set of inputs contained  $\theta_1, \theta_2, \alpha_1, \alpha_2, \omega_1, \omega_2, \theta_{03}, \theta_{04}, \alpha_3, \alpha_4, \omega_3$  and  $\omega_4$ . TEST.M produced consistent

accelerometer values according to the given inputs. These values were then used as inputs to KINET.M to determine  $\theta_{03}$ ,  $\theta_{04}$ ,  $\alpha_3$ ,  $\alpha_4$ ,  $\omega_3$  and  $\omega_4$ . Using the same  $\theta_1$ ,  $\theta_2$ ,  $\alpha_1$ ,  $\alpha_2$ ,  $\omega_1$  and  $\omega_2$  and the accelerometer values from TEST.M, KINET.M produced identical  $\theta_{03}$ ,  $\theta_{04}$ ,  $\alpha_3$ ,  $\alpha_4$ ,  $\omega_3$  and  $\omega_4$  to the inputs to TEST.M.

In TEST.M  ${}^0\mathbf{a}_2$  was first determined using Equation 3.1, which was then used to determine  ${}^3\mathbf{a}_2$  using Equation 4.1.

$$\begin{bmatrix} {}^3a_{2x} \\ {}^3a_{2y} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{03}) & -\sin(\theta_{03}) \\ \sin(\theta_{03}) & \cos(\theta_{03}) \end{bmatrix}^{-1} \begin{bmatrix} {}^0a_{2x} \\ {}^0a_{2y} \end{bmatrix} \quad (4.1)$$

The following two equations are used to determine  ${}^3\mathbf{a}_3$ ,

$${}^3a_{3x} = {}^3a_{2x} + {}^3r_{32y}\alpha_3 + {}^3r_{32x}\omega_3^2 \quad (4.2)$$

$${}^3a_{3y} = {}^3a_{2y} - {}^3r_{32x}\alpha_3 - {}^3r_{32y}\omega_3^2 \quad (4.3)$$

where  ${}^3\mathbf{r}_{32}$  stands for the position vector of the foot joint w.r.t Frame 3.

Finally the theoretical values of the accelerometers attached to the calf were determined as follows.

$${}^3\mathbf{a}_{c1} = {}^3\mathbf{a}_3 + \begin{bmatrix} -{}^3r_{c1y} & -{}^3r_{c1x} \\ {}^3r_{c1x} & {}^3r_{c1y} \end{bmatrix} \begin{bmatrix} \alpha_3 \\ \omega_3^2 \end{bmatrix} \quad (4.4)$$

$${}^3\mathbf{a}_{c2} = {}^3\mathbf{a}_3 + \begin{bmatrix} -{}^3r_{c2y} & -{}^3r_{c2x} \\ {}^3r_{c2x} & {}^3r_{c2y} \end{bmatrix} \begin{bmatrix} \alpha_3 \\ \omega_3^2 \end{bmatrix} \quad (4.5)$$

where  ${}^3\mathbf{r}_{c1}$  and  ${}^3\mathbf{r}_{c2}$  represent the position vectors of accelerometers attached to the calf with

respect to Frame 3.

Since gravity exerts a  $-g$  acceleration on accelerometers, the theoretical accelerometer values were converted to their respective measurements as follows (Equations 4.6 and 4.7).

$${}^3a_{c1g} = {}^3a_{c1} - \begin{bmatrix} g\sin(\theta_{03}) \\ g\cos(\theta_{03}) \end{bmatrix} = a_{c1} \quad (4.6)$$

$${}^3a_{c2g} = {}^3a_{c2} - \begin{bmatrix} g\sin(\theta_{03}) \\ g\cos(\theta_{03}) \end{bmatrix} = a_{c2} \quad (4.7)$$

For the thigh section,  ${}^0a_3$  was first obtained using Equation 4.8.

$$\begin{bmatrix} {}^0a_{3x} \\ {}^0a_{3y} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{03}) & -\sin(\theta_{03}) \\ \sin(\theta_{03}) & \cos(\theta_{03}) \end{bmatrix} \begin{bmatrix} {}^3a_{3x} \\ {}^3a_{3y} \end{bmatrix} \quad (4.8)$$

This procedure was repeated for the calf section to produce the measurements of the accelerometers attached to the thigh section,  $a_{t1}$  and  $a_{t2}$ .

The inputs to TEST.M, the consistent accelerometer values produced by TEST.M. and the output of KINET.M are shown in Appendix A.1-A.3, respectively. The first six rows in Appendix A.1-A.3 are all the same and represent the designed  $\theta_1$ ,  $\omega_1$ ,  $\alpha_1$ ,  $\theta_2$ ,  $\omega_2$ , and  $\alpha_2$ , respectively. The last four rows in Appendix A.1 are the designed  $\theta_{03}$ ,  $\omega_3$ ,  $\alpha_3$ ,  $\theta_{04}$ ,  $\omega_4$  and  $\alpha_4$ .

Accelerometer values produced by TEST.M are the last eight rows representing  $a_{c1x}$ ,  $a_{c1y}$ ,  $a_{c2x}$ ,  $a_{c2y}$ ,  $a_{t1x}$ ,  $a_{t1y}$ ,  $a_{t2x}$  and  $a_{t2y}$ , respectively. The results of simulation shown in Appendix A.3 reveal that the kinematics program works perfectly.

## CONCLUSIONS

A two dimensional analytical model to determine the kinematic parameters of a subject bicycling was developed based on a previously specified measurement system. It was implemented in a MATLAB program, and then proved to work by simulation. This system and method may be used to determine the angular velocity and angular acceleration, linear acceleration, position and orientation of a moving object, knowing the acceleration and position of a known point in the object with respect to the base frame. The precision error of accelerometer values results in an error of the rotation angle between the inertial frame and the moving frame attached to the lower limbs. A theoretical analysis of the resultant error offers an estimate and methods to reduce the error. Reduction of the error can be achieved by increasing the distance between the accelerometers. The results of the numerical error calculations based on the simplified five-bar linkage model are shown in Figures 3.4 and 3.5. These figures show that the errors did decrease with larger distance between the accelerometers. The maximum errors corresponding to 5% accelerometer precision error were determined to be about 6 and 8.25 degrees for different separations between accelerometers; and the maximum errors corresponding to 3% accelerometer precision error were determined to be about 3.6 and 5 degrees.

The A/D board and the decoder board have been proven to work well. The accelerometers are not yet fully usable, and that is why an experimental verification is outside the objective of this thesis. The encoders have been tested and work well. The five D/A



programs have also been proven to work well. The fastest rates of data acquisition of the programs SCANOPF.CPP, SCANOPF2.CPP, SCANOPB.CPP, SCANOPB2.CPP and SCANSTAR.CPP are 57 Hz, 63 Hz, 65 Hz, 67 Hz and above 1.2 kHz, respectively. A complete experimental verification of the proposed system cannot be realized until the hardware is ready.

To quantify bone loading during exercise, forces exerted by a particular muscle, or muscle group, and forces and torques at joints need to be determined. The measurement system can be extended to include load cells to measure contact forces, and EMG probes to provide qualitative information about muscle action. Using these additional sensors, and optimization techniques, it is possible to determine musculoskeletal loading.

## **RECOMMENDATIONS**

Because of the need to synchronize with the decoder board's data acquisition operation, a large drop of the rate of data acquisition occurs. The drop should be greatly reduced if the A/D board can directly acquire the encoders' data. Usually, encoder data includes two chains of TTL pulses which are decoded by the decoder board. Therefore, a program needs to be developed to interpret encoder data that would be digitized using the A/D board (Philippe Coiffet, and Michel Chirouze, 1982).

The model for determining kinematic parameters in this research only works for two dimensions. This model should be generalized to the three dimensional case. The generalization is left for future work.

## LITERATURE CITED

- Abdel-Rahman, E., and Hefzy, M. S.(1993), "Three-Dimensional dynamic modeling of the tibio-femoral joint." *Advances in Bioengineering, ASME Winter Annual Meeting. BED-Vol. 26*, pp. 315-318.
- Anderson, F. C., Ziegler, J. M., Pandy M. G., and Whalen, R. T. (1993), "Numerical computation of optimal controls for large-scale musculoskeletal systems." *Advances in Bioengineering, ASME Winter Annual Meeting. BED-Vol. 26*, pp. 519-522.
- Ericson, M. O., Ekholm, J., Svensson, O., and Nisell, R. (1985), "The forces of ankle joint structures during ergometer cycling." *J. of the American Orthopaedic Foot and Ankle Soc.*, Vol. 6, No. 3, pp. 135-142.
- Fernando Figueroa, "Loading, Electromyograph, and Motion During Exercise." Final Report, NASA/ASEE Summer Faculty Fellowship Program, Johnson Space Center, 1993.
- Harrison, R. N., Lees, A., McCullagh, P. J. J., and Rowe, W. B. (1986), "A bioengineering analysis of human muscle and joint forces in the lower limbs during running." *J. of Sports Sciences*, Vol. 4, pp. 201-218.
- Phillip John Mckerrow, *Introduction to Robotics*, pp. 137-148.
- Philippe Coiffet, and Michel Chirouze. (1982), *An introduction to ROBOT TECHNOLOGY*, pp.111-113
- Redfield, R., and Hull, M. L. (1986A), "Prediction of pedal forces in bicycling using optimization meghods." *J. Biomechanics*. Vol. 19, No. 7, pp. 523-540.

- Redfield, R., and Hull, M. L. (1986B), "On the relation between joint moments and pedalling rates at constant rates at constant power in bicycling." *J. Biomechanics*, Vol. 19, No. 4, pp. 317-329.
- Verstraete, M. C. (1992), "A technique for locating the center of mass and principal axes of the lower limb." *Official Journal of the American College of Sports Med.* Special Communications, pp. 825-831.
- Yang, Y., Yahia, L. H., and Feldman, A. G. (1993), "A versatile dynamic model of human arm." *Advances in Bioengineering, ASME Winter Annual Meeting. BED-Vol. 26*, pp. 527-529..
- Y. C. Fung, (1919). *Biomechanics, Mechanical Properties of Living Tissues*, pp.393-396.
- NI-DAQ Function Reference Manual, p3-165, p3-169, p3-33, p3-47.
- Turbo C++ User's Guide, p579.

## APPENDIX A.1 INPUTS TO TEST.M

input =

Columns 1 through 7

10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	30.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
0	0	0	0	0	0	0
30.0000	30.0000	30.0000	30.0000	30.0000	30.0000	30.0000
3.0000	2.0000	1.0000	3.0000	2.0000	1.0000	3.0000
1.5000	0	0	1.5000	0	0	1.5000
4.0000	1.0000	2.0000	4.0000	1.0000	2.0000	4.0000
2.5000	3.0000	7.7330	2.5000	3.0000	7.7330	2.5000
45.0000	50.0000	55.0000	60.0000	65.0000	70.0000	75.0000
2.0000	3.0000	4.0000	2.0000	3.0000	4.0000	2.0000
1.0000	7.0000	9.0000	1.0000	7.0000	9.0000	1.0000
135.0000	140.0000	145.0000	150.0000	155.0000	160.0000	165.0000

Columns 8 through 14

30.0000	30.0000	30.0000	30.0000	30.0000	30.0000	30.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
0	0	0	0	0	0	0
30.0000	30.0000	30.0000	30.0000	30.0000	30.0000	30.0000
2.0000	1.0000	3.0000	2.0000	1.0000	3.0000	2.0000
0	0	1.5000	0	0	1.5000	0
1.0000	2.0000	4.0000	1.0000	2.0000	4.0000	1.0000
3.0000	7.7330	2.5000	3.0000	7.7330	2.5000	3.0000
80.0000	85.0000	90.0000	95.0000	100.0000	105.0000	110.0000
3.0000	4.0000	2.0000	3.0000	4.0000	2.0000	3.0000
7.0000	9.0000	1.0000	7.0000	9.0000	1.0000	7.0000
170.0000	175.0000	180.0000	185.0000	190.0000	195.0000	200.0000

Columns 15 through 19

30.0000	30.0000	30.0000	30.0000	30.0000
2.0000	2.0000	2.0000	2.0000	2.0000
0	0	0	0	0
30.0000	30.0000	30.0000	30.0000	30.0000
1.0000	3.0000	2.0000	1.0000	1.0000
0	1.5000	0	0	0
2.0000	4.0000	1.0000	2.0000	2.0000
7.7330	2.5000	3.0000	7.7330	7.7330
115.0000	120.0000	125.0000	130.0000	135.0000
4.0000	2.0000	3.0000	4.0000	4.0000
9.0000	1.0000	7.0000	9.0000	9.0000
205.0000	210.0000	215.0000	220.0000	225.0000

## APPENDIX A.2 ACCELEROMETER VALUES PRODUCED BY TEST.M

accelerometer =

1.0e+003 \*

Columns 1 through 7

0.0002	0.0002	0.0002	0.0002	0.0002	0.0002	0.0005
0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020
0	0	0	0	0	0	0
0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
0.0030	0.0020	0.0010	0.0030	0.0020	0.0010	0.0030
0.0015	0	0	0.0015	0	0	0.0015
-1.6468	-1.1093	-1.1135	-1.7509	-1.2010	-1.1891	-1.8695
-0.4026	-0.4033	-0.1762	-0.0784	-0.1031	0.1056	0.1191
-1.3268	-1.0893	-1.0335	-1.4309	-1.1810	-1.1091	-1.5495
-0.4526	-0.4633	-0.3308	-0.1284	-0.1631	-0.0491	0.0691
-0.5726	-0.6133	-0.5538	-0.2484	-0.3131	-0.2721	-0.0509
1.8693	1.3893	1.5422	1.9734	1.4810	1.6178	2.0920
-0.4926	-0.4333	-0.2338	-0.1684	-0.1331	0.0479	0.0291
1.8493	1.2493	1.3622	1.9534	1.3410	1.4378	2.0720

Columns 8 through 14

0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
0.0020	0.0020	0.0020	0.0020	0.0020	0.0020	0.0020
0	0	0	0	0	0	0
0.0005	0.0005	0.0005	0.0005	0.0005	0.0005	0.0005
0.0020	0.0010	0.0030	0.0020	0.0010	0.0030	0.0020
0	0	0.0015	0	0	0.0015	0
-1.2956	-1.2603	-1.8310	-1.2470	-1.1977	-1.6996	-1.1145
0.1142	0.3383	0.4832	0.4450	0.6405	0.8249	0.7519
-1.2756	-1.1803	-1.5110	-1.2270	-1.1177	-1.3796	-1.0945
0.0542	0.1837	0.4332	0.3850	0.4858	0.7749	0.6919
-0.0958	-0.0394	0.3132	0.2350	0.2628	0.6549	0.5419
1.5756	1.6889	2.0535	1.5270	1.6264	1.9221	1.3945
0.0842	0.2806	0.3932	0.4150	0.5828	0.7349	0.7219
1.4356	1.5089	2.0335	1.3870	1.4464	1.9021	1.2545

Columns 15 through 19

0.0005	0.0005	0.0005	0.0005	0.0005
0.0020	0.0020	0.0020	0.0020	0.0020
0	0	0	0	0
0.0005	0.0005	0.0005	0.0005	0.0005
0.0010	0.0030	0.0020	0.0010	0.0010
0	0.0015	0	0	0
-1.0592	-1.4842	-0.9070	-0.8539	-0.7730
0.9162	1.1210	1.0141	1.1467	1.2106
-0.9792	-1.1642	-0.8870	-0.7739	-0.6930
0.7615	1.0710	0.9541	0.9920	1.0559
0.5385	0.9510	0.8041	0.7690	0.8329
1.4678	1.7067	1.1870	1.2826	1.2017
0.8585	1.0310	0.9841	1.0890	1.1529
1.3078	1.6867	1.0470	1.1026	1.0217



## APPENDIX A.3 THE RESULTS OF SIMULATION

simulation =

Columns 1 through 7

10.0000	10.0000	10.0000	10.0000	10.0000	10.0000	30.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
0	0	0	0	0	0	0
30.0000	30.0000	30.0000	30.0000	30.0000	30.0000	30.0000
3.0000	2.0000	1.0000	3.0000	2.0000	1.0000	3.0000
1.5000	0	0	1.5000	0	0	1.5000
4.0000	1.0000	2.0000	4.0000	1.0000	2.0000	4.0000
2.5000	3.0000	7.7330	2.5000	3.0000	7.7330	2.5000
45.0000	50.0000	55.0000	60.0000	65.0000	70.0000	75.0000
2.0000	3.0000	4.0000	2.0000	3.0000	4.0000	2.0000
1.0000	7.0000	9.0000	1.0000	7.0000	9.0000	1.0000
135.0000	140.0000	145.0000	150.0000	155.0000	160.0000	165.0000

Columns 8 through 14

30.0000	30.0000	30.0000	30.0000	30.0000	30.0000	30.0000
2.0000	2.0000	2.0000	2.0000	2.0000	2.0000	2.0000
0	0	0	0	0	0	0
30.0000	30.0000	30.0000	30.0000	30.0000	30.0000	30.0000
2.0000	1.0000	3.0000	2.0000	1.0000	3.0000	2.0000
0	0	1.5000	0	0	1.5000	0
1.0000	2.0000	4.0000	1.0000	2.0000	4.0000	1.0000
3.0000	7.7330	2.5000	3.0000	7.7330	2.5000	3.0000
80.0000	85.0000	90.0000	95.0000	100.0000	105.0000	110.0000
3.0000	4.0000	2.0000	3.0000	4.0000	2.0000	3.0000
7.0000	9.0000	1.0000	7.0000	9.0000	1.0000	7.0000
170.0000	175.0000	180.0000	185.0000	190.0000	195.0000	200.0000

Columns 15 through 19

30.0000	30.0000	30.0000	30.0000	30.0000
2.0000	2.0000	2.0000	2.0000	2.0000
0	0	0	0	0
30.0000	30.0000	30.0000	30.0000	30.0000
1.0000	3.0000	2.0000	1.0000	1.0000
0	1.5000	0	0	0
2.0000	4.0000	1.0000	2.0000	2.0000
7.7330	2.5000	3.0000	7.7330	7.7330
115.0000	120.0000	125.0000	130.0000	135.0000
4.0000	2.0000	3.0000	4.0000	4.0000
9.0000	1.0000	7.0000	9.0000	9.0000
205.0000	210.0000	215.0000	220.0000	225.0000

»

## APPENDIX B.1 KINET.M

```

% *****
% set the parameters in file and read from it to determine variable

% *****

```

```

function x=kinet(kar,b);

```

```

% b is a scale factor

```

```

% define some constant variable

```

```

l1= 15*b ; % crank length
l2= 10*b ; % the length between foot joint and
           % pedal

```

```

l3= 40*b ; % calf
l4= 40*b ; % thigh

```

```

% % % % % % % % % for left pedal

```

```

rc1=[ -10,-5]*b ; % in frame 3
rc2=[ -30,-5]*b ; % in frame 3
rt1=[ -10,-5]*b ; % in frame 4
rt2=[ -30,-5]*b ; % in frame 4
r32=[ -l3 0]';
r43=[ -l4 0]';

```

```

% for right pedal

```

```

% rc1=[ -2,-2]*b ; % in frame 3
% rc2=[ -4,-2]*b ; % in frame 3
% rt1=[ -2,-2]*b ; % in frame 4
% rt2=[ -4,-2]*b ; % in frame 4
% r32=[ -l3 0]';
% r43=[ -l4 0]';

```

```

g=980*b;

```

```
% get paremetesr from data acquisition program
```

```
[m,n]= size(kar);
```

```
crank(1,:)=kar(1,:);
```

```
w_c(1,:)=kar(2,:);
```

```
arfa_c(1,:)=kar(3,:);
```

```
%%%%%%%% loop for kinetmatics %%%%%%%%%
```

```
for j=1:2    % 2 for right pedal, position vectors need to be redefined if not the same as  
            % the left
```

```
if j==1,    %%%%%%%%%for left pedal
```

```
pedal(1,:)=kar(4,:);
```

```
w_p(1,:)=kar(5,:);
```

```
arfa_p(1,:)=kar(6,:);
```

```
ac1(1:2,:)=kar(7:8,:);
```

```
at1(1:2,:)=kar(11:12,:);
```

```
ac2(1:2,:)=kar(9:10,:);
```

```
at2(1:2,:)=kar(13:14,:);
```

```
else        %%%%%%%%%for right pedal
```

```
pedal(1,:)=kar(15,:);
```

```
for i=1:n
```

```
crank(1,i)=crank(1,i)+pi;
```

```
end
```

```
w_p(1,:)=kar(16,:);
```

```
arfa_p(1,:)=kar(17,:);
```

```
ac1(1:2,:)=kar(18:19,:);
```

```
at1(1:2,:)=kar(22:23,:);
```

```
ac2(1:2,:)=kar(20:21,:);
```

```
at2(1:2,:)=kar(24:25,:);
```

```
end
```

```
rc12=rc2-rc1;
```

```
rt12=rt2-rt1;
```

```
for i=1:n
```

```
    ac12=ac2(:,i)-ac1(:,i);  
    at12=at2(:,i)-at1(:,i);
```

```
    S1=sin(crank(1,i));  
    C1=cos(crank(1,i));
```

```
    % determine a02 in frame 0
```

```
    S12=sin(crank(1,i)+pedal(1,i));  
    C12=cos(crank(1,i)+pedal(1,i));  
    t3=[ -l1*S1 -l2*S12; l1*C1 l2*C12];
```

```
    t4=[ arfa_c(1,i) (arfa_c(1,i)+arfa_p(1,i))]';
```

```
    t5=[ l1*C1 l2*C12;  
         l1*S1 l2*S12];
```

```
    t6=[ (w_c(1,i)^2) (w_c(1,i)+w_p(1,i))^2]';
```

```
    a02(1:2,i)=t3*t4-t5*t6; % in frame 0
```

```
    % determine a03 in frame3
```

```
    t1=inv(arr(rc12))*ac12;  
    w_ca(1,i)=sqrt(t1(2,1));  
    arfa_ca(1,i)=t1(1,1);  
    t2=arr(rc1)*t1;  
    a03_3(1:2,i)=-t2+ac1(:,i); % in frame 3
```

```
    % determine rotation angle between frame 0 and frame 3
```

```

a02_3=a03_3(:,i)+arr(r32)*t1;      % in frame 3 and without compensation of
                                   % gravity
                                   % a02_3 should be a02_3+com(g,theat03)

t8(i)=(a02(1,i)*a02_3(1,1)+a02(2,i)*a02_3(2,1)-a02_3(2,1)*g)/((a02_3(1,1)^2)+(a02_3(2,1)
    ^2));

t9=(a02(2,i)*a02_3(1,1)-a02(1,i)*a02_3(2,1)-a02_3(1,1)*g)/((a02_3(1,1)^2)+(a02_3(2,1)^2))
;

theta03(1,i)=acos(t8(i));

a03_3(1:2,i)=a03_3(1:2,i)+commm(g,theta03(:,i)); % COMPENSATION WITH
                                                    %GRAVITY;

a03(1:2,i)=(rot2d(theta03(:,i)))*a03_3(:,i);    % in frame 0;

% % % % % % % determine a04      in frame4 % % % % % % % % % % % % % % % % % % % % % % %

t1=inv(arr(rt12))*at12;
w_t(1,i)=sqrt(t1(2,1));
arfa_t(1,i)=t1(1,1);
t2=arr(rt1)*t1;

a04_4(:,i)=-t2+at1(:,i);      % in frame 4

% determine rotation angle between frame 0 and frame 4
a03_4=a04_4(:,i)+arr(r43)*t1;      % in frame 3 and without compensation of
                                   % gravity
                                   % it should be a03_4+com(g,theta04)

theta04(1,i)=acos((a03(1,i)*a03_4(1,1)+a03(2,i)*a03_4(2,1)-a03_4(2,1)*g)/((a03_4(1,1)^2)
    +(a03_4(2,1)^2)));

t9=(a03(2,i)*a03_4(1,1)-a03(1,i)*a03_4(2,1)-a03_4(1,1)*g)/((a03_4(1,1)^2)+(a03_4(2,1)^2))
;

```

```
if t9 <= 0
```

```
    theta04(1,i)=pi+asin(-t9);  
end
```

```
    a04_4(:,i)=a04_4(:,i)+commm(g,theta04(:,i));           % COMPENSATION WITH  
                                                            % GRAVITY
```

```
    a04(:,i)=(rot2d(theta04(:,i)))*a04_4(:,i);
```

```
    end % end of operation loop
```

```
if j==1
```

```
    a03_l(:,:)=a03(:,:);  
    a04_l(:,:)=a04(:,:);  
    theta03_l(1,:)=theta03(1,:)*180/pi;  
    theta04_l(1,:)=theta04(1,:)*180/pi;  
    arfa_cal(1,:)=arfa_ca(1,:);  
    w_cal(1,:)=w_ca(1,:);  
    arfa_tl(1,:)=arfa_t(1,:);  
    w_tl(1,:)=w_t(1,:);
```

```
else
```

```
    a03_r(:,:)=a03(:,:);  
    a04_r(:,:)=a04(:,:);  
    theta03_r(1,:)=theta03(1,:)/pi*180;  
    theta04_r(1,:)=theta04(1,:)/pi*180;  
    arfa_car(1,:)=arfa_ca(1,:);  
    w_car(1,:)=w_ca(1,:);  
    arfa_tr(1,:)=arfa_t(1,:);  
    w_tr(1,:)=w_t(1,:);
```

```
end % end of if
```

```
end % end for j
```



```
x(1,:)=w_cal(1,:);  
x(2,:)=arfa_cal(1,:);  
x(3,:)=theta03_l(1,:)*180/pi;  
x(4,:)=w_tl(1,:);  
x(5,:)=arfa_tl(1,:);  
x(6,:)=theta04_l(1,:)*180/pi;  
x(7:8,:)=a03_l(:,,:);  
x(9:10,:)=a04_l(:,,:);
```

```
x(11,:)=w_car(1,:);  
x(12,:)=arfa_car(1,:);  
x(13,:)=theta03_r(1,:)*180/pi;  
x(14,:)=w_tr(1,:);  
x(15,:)=arfa_tr(1,:);  
x(16,:)=theta04_r(1,:)*180/pi;  
x(17:18,:)=a03_r(:,,:);  
x(19:20,:)=a04_r(:,,:);
```

## APPENDIX B.2 TEST.M

```

function l=test(kar,b);

% define some constant variable

l1=    15*b    ; % crank length
l2=    10*b    ; % the length between foot joint and
                % pedal

l3=    40*b    ; % calf
l4=    40*b    ; % thigh
%% %% %% %% %% for left pedal
rc1=[ -10,-5]*b ; % in frame 3
rc2=[ -30,-5]*b ; % in frame 3
rt1=[ -10,-5]*b ; % in frame 4
rt2=[ -30,-5]*b ; % in frame 4
r32=[ -l3 0]';
r43=[ -l4 0]';

g=980*b;

% get paremetesr from data acquisition program

[m,n]=size(kar);

crank(1,:)=kar(1,:)*pi/180;
w_c(1,:)=kar(2,:);
arfa_c(1,:)=kar(3,:);
pedal(1,:)=kar(4,:)*pi/180;
w_p(1,:)=kar(5,:);
arfa_p(1,:)=kar(6,:);
w_ca(1,:)=kar(7,:);
theta03(1,:)=kar(9,:)*pi/180;
arfa_ca(1,:)=kar(8,:);
w_t(1,:)=kar(10,:);
arfa_t(1,:)=kar(11,:);
theta04(1,:)=kar(12,:)*pi/180;

for i=1:n

```

```

% determine a01 in frame 0
S1=sin(crank(1,i));
C1=cos(crank(1,i));

% determine a02 in frame 0
S12=sin(crank(1,i)+pedal(1,i));
C12=cos(crank(1,i)+pedal(1,i));
t3=[ -l1*S1 -l2*S12; l1*C1 l2*C12];

t4=[ arfa_c(1,i) (arfa_c(1,i)+arfa_p(1,i))'];

t5=[ l1*C1 l2*C12;
l1*S1 l2*S12];

t6=[ (w_c(1,i)^2) (w_c(1,i)+w_p(1,i))^2]';

a02(1:2,i)=t3*t4-t5*t6; % in frame 0

k=[arfa_ca(1,i) w_ca(1,i)^2]';

a02_3(1:2,i)=inv(rot2d(theta03(1,i)))*a02(:,i);
a03_3(1:2,i)=a02_3(1:2,i)-arr(r32)*k;
ac1(:,i)=a03_3(:,i)+arr(rc1)*k-commm(g,theta03(:,i));
ac2(:,i)=a03_3(:,i)+arr(rc2)*k-commm(g,theta03(:,i));
a03(:,i)=rot2d(theta03(1,i))*a03_3(:,i);
k=[arfa_t(1,i) w_t(1,i)^2]';
a03_4(1:2,i)=inv(rot2d(theta04(1,i)))*a03(:,i);
a04_4(1:2,i)=a03_4(1:2,i)-arr(r43)*k;
at1(:,i)=a04_4(:,i)+arr(rt1)*k-commm(g,theta04(:,i));
at2(:,i)=a04_4(:,i)+arr(rt2)*k-commm(g,theta04(:,i));

end

l(1,:)=crank(1,:);
l(2,:)=w_c(1,:);
l(3,:)=arfa_c(1,:);
l(4,:)=pedal(1,:);
l(5,:)=w_p(1,:);
l(6,:)=arfa_p(1,:);
l(7:8,:)=ac1(1:2,:);
l(9:10,:)=ac2(1:2,:);
l(11:12,:)=at1(1:2,:);
l(13:14,:)=at2(1:2,:);

```